

Universidad Complutense de Madrid

Facultad de Informática



**Control hardware de una FPGA para el estudio
de su comportamiento bajo radiación en
entornos hostiles**

Autor David Ortiz Fernández

**Directores de proyecto Hortensia Mecha López
Juan Carlos Fabero Jiménez**

Trabajo de Fin de Grado

Grado en Ingeniería de Computadores

Junio de 2018

Agradecimientos

Escribir este trabajo ha supuesto un gran enriquecimiento en mis conocimientos y es por eso por lo que me gustaría agradecer a todas aquellas personas que me han ayudado y apoyado durante este proceso.

Primero de todo, me gustaría agradecer a mi pareja por su apoyo incondicional. Me has apoyado enormemente y siempre has estado ahí para ayudarme cuando te necesitaba.

También me gustaría agradecer a mis padres y a mi hermana por sus sabios consejos y su comprensión. Siempre habéis estado ahí para mí.

Además, me gustaría agradecer a mis tutores Hortensia y Juan Carlos, por su valiosa ayuda e infinita paciencia. Definitivamente me habéis brindado todas las herramientas necesarias para completar mi trabajo de fin de grado satisfactoriamente, no puedo imaginar mejores tutores.

¡Muchas gracias a todos!

Resumen

En la actualidad, los sistemas embebidos han logrado un gran auge gracias a sus diferentes campos de aplicación y a sus bajos costos comparados con sistemas de cómputo tradicionales. Desde su invención en el año 1984, la gran capacidad de reconfiguración y adaptación que ofrecen las FPGA las hace muy atractivas en múltiples sectores, tales como el aeroespacial o el industrial. Sin embargo, los efectos de la radiación en estos dispositivos, cuyo comportamiento viene determinado por una memoria de configuración, pueden ser tan catastróficos que su uso en entornos de alta radiación debe venir acompañado por una serie de técnicas de protección, tanto a nivel tecnológico como de diseño. Por este motivo, y para realizar una posterior investigación sobre el comportamiento de un determinado circuito implementado en una FPGA en entornos hostiles y bajo las condiciones mencionadas, surge este trabajo. Para realizar las pruebas de inyección de fallos, bien a través de plataformas de emulación o de un sistema de aceleración de partículas, es necesario un sistema externo de control del circuito, en particular de los datos de entrada-salida y del reloj del mismo. Este sistema de generación de reloj, que se ha desarrollado en este trabajo, se ha integrado en un sistema global de inyección de errores que permite poner a prueba la vulnerabilidad de cualquier circuito frente a radiación real. En este trabajo se aborda el estudio de la transmisión de una señal de reloj de manera fiable entre dos FPGA a distancia moderada, mediante la implementación de diferentes técnicas, que permiten aumentar la inmunización al ruido impulsivo y la radiación electromagnética. Para alcanzar este propósito se han utilizado varias FPGA de las familias Artix 7 y Virtex 5 del fabricante Xilinx. Por último, se han obtenido y comparado resultados reales del envío de dicha señal a diferentes distancias y frecuencias.

Abstract

Currently embedded systems have achieved a great boom thanks to its different fields of application and its low costs compared to traditional computer systems. Since its invention in 1984, the great capacity of reconfiguration and adaptation offered by FPGAs makes them very attractive in multiple sectors such as aerospace or industrial. However, the effects of radiation on these devices, whose behavior is determined by a configuration memory can be so catastrophic that its use in high radiation environments must be accompanied by a series of technological and design protection techniques. . For this reason and to carry out a subsequent investigation on the behavior of a specific circuit implemented in an FPGA in hostile environments and under the conditions mentioned, the reason for this project arises. To perform the fault injection tests either through emulation platforms or a particle acceleration system, an external circuit control system is needed, in particular the input-output data and its clock. This clock generation system will be integrated into a global error injection system that will allow testing the vulnerability of any circuit against real radiation. The objective of this work is to study the transmission of a clock signal in a reliable way between two FPGAs through the implementation of different techniques, through immunization to impulsive noise and electromagnetic radiation are addressed. To achieve this purpose, several FPGAs from the Artix 7 and Virtex 5 families of the manufacturer Xilinx have been used. Finally, real results of the sending of said signal at different distances and frequencies have been obtained and compared.

Índice general

| | |
|--|-----------|
| Agradecimientos | 3 |
| Resumen | 5 |
| Lista de figuras | 11 |
| Lista de tablas | 13 |
| Palabras clave | 15 |
| Acrónimos | 17 |
| 1. Introducción | 19 |
| 1.1. Sistemas embebidos y FPGA | 19 |
| 1.2. Motivaciones | 19 |
| 1.3. Ruido en la transmisión | 21 |
| 1.4. Objetivos de este proyecto | 22 |
| 1.5. Fases del trabajo | 23 |
| 2. Entorno de desarrollo | 25 |
| 2.1. Entorno <i>hardware</i> | 25 |
| 2.1.1. Placa de desarrollo Basys 3 | 25 |
| 2.1.1.1. UART | 26 |
| 2.1.1.2. JTAG | 27 |
| 2.1.1.3. Conectores PMOD de la placa Basys 3 | 28 |
| 2.1.2. Placa de desarrollo XUPV5-LX110T | 28 |
| 2.1.3. FPGA | 29 |
| 2.1.3.1. Arquitectura Xilinx Artix 7 | 30 |
| 2.1.3.2. Arquitectura Xilinx Virtex 5 | 33 |
| 2.1.3.3. Arquitectura de reloj en las FPGA de Xilinx | 33 |
| 2.2. Herramientas <i>software</i> para el desarrollo | 35 |
| 2.2.1. Vivado Design Suite | 35 |
| 2.2.2. Xilinx EDK | 36 |
| 2.2.3. Xilinx SDK | 36 |

| | |
|---|-----------|
| 2.2.4. Impact | 37 |
| 3. Hardware para transmisión fiable | 39 |
| 3.1. Arquitectura <i>Hardware</i> | 40 |
| 3.1.1. MicroBlaze | 40 |
| 3.1.2. Bus AXI | 40 |
| 3.1.3. Bus LMB | 41 |
| 3.1.4. Adaptador UART AXI | 42 |
| 3.1.5. Asistente de sincronización | 42 |
| 3.1.6. Circuito controlador de ciclos de reloj | 43 |
| 3.1.7. Circuito bajo prueba | 44 |
| 3.1.8. Sistema <i>Software</i> | 44 |
| 4. Metodología | 45 |
| 4.1. Introducción a la metodología | 45 |
| 4.2. Optimización de los elementos de transmisión pasivos. | 46 |
| 4.2.1. Diafonía. Trenzado de cables | 47 |
| 4.2.1.1. Atenuación | 48 |
| 4.2.1.2. AWG | 48 |
| 4.3. Mejoras sobre elementos activos. Señales diferenciales | 48 |
| 4.3.1. Voltajes asociados a los IOB | 49 |
| 4.3.2. Buffers diferenciales. Amplificadores operacionales | 49 |
| 4.3.2.1. OBUFDS | 50 |
| 4.3.2.2. IBUFDS | 51 |
| 4.3.3. Terminación del cable | 51 |
| 4.3.3.1. Señal reflejada | 53 |
| 4.3.3.2. Resistencia de terminación | 53 |
| 4.4. Efectos jitter, skew y slew rate | 54 |
| 4.5. Limitaciones encontradas | 55 |
| 4.5.1. Conectores SMA. Placa XUPV5-LX110T | 56 |
| 5. Resultados | 57 |
| 5.1. Resultados del diseño original | 57 |
| 5.2. Resultados con señales diferenciales | 58 |
| 5.3. Resultados añadiendo una resistencia de terminación | 58 |
| 5.4. Resultados usando placas XUPV5-LX110T y conectores SMA | 59 |
| 6. Conclusiones | 61 |
| 7. Futuras líneas de investigación | 65 |
| Bibliografía | 68 |

Índice de figuras

| | |
|---|----|
| 1.1. Comportamiento del sustrato silicio ante la incidencia de un neutrón de alta velocidad. Figura cortesía de [1] | 21 |
| 1.2. Diagrama de la plataforma desarrollada. | 21 |
| 2.1. FPGA Artix 7 empotrada en placa Basys 3. | 26 |
| 2.2. Conexión entre el <i>chip</i> FT2232HQ y Artix 7. Figura cortesía de [2]. | 27 |
| 2.3. Interfaz JTAG conectada en <i>daisy-chain</i> | 27 |
| 2.4. FPGA Virtex 5 empotrada en placa XUPV5-LX110T. | 28 |
| 2.5. Circuito integrado destinado al control de los conectores SMA de salida de reloj. Figura cortesía de [3]. | 29 |
| 2.6. Arquitectura básica de una FPGA. | 30 |
| 2.7. SLICEL perteneciente a un CLB de la familia Artix 7. Figura cortesía de [4]. . . | 31 |
| 2.8. Distribución de los IOB en la FPGA XC7A35T 1CPG236C. Figura cortesía de [5]. | 32 |
| 2.9. Diagrama de bloques de un CMT de la serie 7 de Xilinx. Figura cortesía de [6]. . | 32 |
| 2.10. Diagrama de bloques de un CMT de la familia Virtex 5 de Xilinx. Figura cortesía de [7]. | 34 |
| 2.11. Arquitectura de reloj de la serie 7 de Xilinx. Figura cortesía de [6]. | 35 |
| 2.12. Distribución de buffers de reloj en la serie 7 de Xilinx. Figura cortesía de [6]. . . | 36 |
| 3.1. Arquitectura <i>hardware</i> de la plataforma que controla la emisión de ciclos de reloj. | 39 |
| 3.2. Arquitectura MicroBlaze. Figura cortesía de [8]. | 41 |
| 3.3. Transacción de escritura AXI. Figura cortesía de [9]. | 42 |
| 3.4. Diagrama de bloques del asistente de sincronización. Figura cortesía de [10]. . . | 43 |
| 3.5. Máquina de estados que controla la emisión de ciclos de reloj. | 44 |
| 4.1. Diagrama que muestra la metodología de trabajo para realizar las mejoras y mediciones. | 45 |
| 4.2. Cable UTP categoría 6. | 47 |
| 4.3. Requisitos VCCO y VREF para los estándares seleccionados. Figura cortesía de [11]. | 49 |
| 4.4. Símbolo del amplificador operacional. | 50 |
| 4.5. Resistencia, inductancia y capacitancia de un cable no ideal. | 52 |
| 4.6. Diseño de terminación para señalización diferencial. Figura cortesía de [12]. . . . | 54 |

4.7. Diseño de terminación propuesto por Xilinx. Figura cortesía de [11]. 54

4.8. Efecto *jitter*. 55

Índice de tablas

| | | |
|------|--|----|
| 4.1. | Características cable <i>Ethernet</i> categoría 6. | 46 |
| 4.2. | Valores de AWG 24. | 48 |
| 5.1. | Resultados en la transmisión de una señal de reloj digital a 100MHz. | 57 |
| 5.2. | Resultados en la transmisión de una señal de reloj en modo diferencial a 100MHz. | 58 |
| 5.3. | Resultados en la transmisión de una señal de reloj en modo diferencial para diferentes frecuencias sobre placas Basys 3. | 59 |
| 5.4. | Resultados en la transmisión de una señal de reloj en modo diferencial a diferentes frecuencias sobre placas XUPV5-LX110T. | 60 |

Palabras clave

Palabras clave en Español

- FPGA
- Hardware reconfigurable
- Inmunización al ruido
- Radiación de partículas
- Reloj diferencial
- Ruido electromagnético
- SEU
- Sistema empotrado
- Transmisión fiable

Keywords in English

- FPGA
- Reconfigurable Hardware
- Immunization to noise
- Particle radiation
- Differential Clock
- Electromagnetic noise
- SEU
- Embedded system
- Reliable transmission

Lista de Acrónimos

AMBA Advanced Microcontroller Bus Architecture

ASIC Application-Specific Integrated Circuits

ASSP Application-Specific Standard Parts

AWG American Wire Gauge

AXI Advanced Extensible Interface

CCIO Clock Capable Input Output

FPGA Field-Programmable Gate Array

HR High Range

IC Integrated Circuit

IOB Input Output Block

IDE Integrated Design Enviroment

JTAG Joint Test Action Group

LMB Local Memory Bus

LUT Look-Up Table

MMCM Mixed-Mode Clock Manager

PCB Printed Circuit Board

PSM Programmable Switch Matrix

PLL Phase Locked Loop

RAM Random Access Memory

RTL Register-Transfer Level

SEU Single Event Upset

SET Single Event Transient

SMA SubMiniature version A

SoC System on Chip

STP Shielded Twisted Pair

TFG Trabajo de Fin de Grado

UART Universal Asynchronous Receiver Transmitter

USB Universal Serial Bus

UTP Unshielded Twisted Pair

Capítulo 1

Introducción

1.1. Sistemas embebidos y FPGA

En las últimas tres décadas los sistemas empotrados han inundado el entorno que nos rodea. Diseñados con el propósito de cubrir multitud de necesidades específicas, son ampliamente utilizados en nuestra vida cotidiana y en diversos sectores tales como el aeroespacial, el aeronáutico, el industrial o el médico. Ya desde la aparición de los primeros sistemas embebidos en los años 80, la NASA (*National Aeronautics and Space Administration*) los empezó a incorporar para el control de los sistemas de guía inercial de los módulos de excursión lunar en las misiones Apolo a la Luna, y desde un principio fueron considerados como los elementos que más riesgo suponían para la ejecución de dichas misiones, debido a que eran susceptibles de una serie de fallos provocados por el entorno a los que estaban destinados [13]. El desarrollo de los sistemas embebidos modernos demanda la implementación de funciones sofisticadas en plazos de diseño cortos. Desde su invención en 1984, los dispositivos FPGA (*Field-Programmable Gate Array*) han ofrecido una gran capacidad de flexibilidad en el diseño e implementación de estos sistemas empotrados frente a los sistemas ASIC (*Application-Specific Integrated Circuit*), a pesar de presentar un mayor consumo de energía y no poder abarcar sistemas tan complejos como ellos. Los dispositivos FPGA permiten la implementación de el *hardware* y *software* de un sistema digital en un circuito integrado configurable, volcándolo el mapa de bits del diseño sobre la memoria de configuración. Muchas FPGA modernas soportan la reconfiguración parcial del sistema, permitiendo que una parte del diseño sea reprogramada, mientras las demás partes siguen funcionando, siendo ésta y las mencionadas anteriormente las principales razones por las que las FPGA son tan atractivas en el sector aeronáutico y aeroespacial.

1.2. Motivaciones

Los rayos cósmicos son partículas, generalmente protones de alta energía, por poseer velocidades cercanas a la de la luz, que se originan en el exterior de nuestra atmósfera, siendo el Sol una de sus principales fuentes junto a las novas y supernovas externas al Sistema Solar. Cuando estos rayos colisionan con las moléculas del aire, se crea una lluvia de partículas en cascada entre las que se incluyen neutrones de alta velocidad. Estos neutrones de alta velocidad son, junto con las

partículas alfa, las principales fuentes de un tipo de cambio no deseado en elementos de memoria, los SEU (*Single Event Upset*), que afecta a dispositivos semiconductores, como es el caso de las FPGA de silicio, cuando impactan sobre ellas [1]. Estas partículas también pueden producir pulsos de corriente transitoria y *latch-up*. De estos efectos, solo el *latch-up* es potencialmente destructivo ya que supone un cortocircuito persistente, que puede ocasionar un sobrecalentamiento, fusión o vaporización en los circuitos electrónicos. Debido a la creciente inclusión de las FPGA en el sector aeroespacial y de la aeronáutica, impulsado en gran medida por el gran aumento en el coste de producción de los sistemas ASIC y ASSP (*Application-Specific Standard Parts*), es importante someter a estudio los efectos de la incidencia de dichas partículas sobre estos dispositivos.

A medida que la atmósfera se vuelve más densa cerca de la superficie de la tierra, hay más moléculas de aire que pueden colisionar con los neutrones de alta velocidad, pero cada colisión produce neutrones de energía cada vez más baja. Por el contrario, a mayores altitudes hay más neutrones de alta velocidad, debido a que hay menos probabilidad de golpear moléculas de aire, aumentando así el recuento de neutrones por área y por tiempo con la altitud [14].

Independientemente del origen, cuando uno de estos neutrones de alta velocidad pasa a través del dispositivo y choca con un átomo en el sustrato de silicio, se produce una reacción que libera una nube de huecos de electrón. Esto crea en el dispositivo un SET (*Single-Event Transients*), que puede propagarse a través de la lógica combinacional. Si el SET tiene suficiente amplitud y duración y coincide con un flanco de reloj, entonces se registra como un valor incorrecto convirtiéndose en un SEU. Un SEU consiste en el cambio de estado de un elemento de almacenamiento dentro de un dispositivo y está producido al colisionar un neutrón con un átomo de silicio o por un contaminante que emite una partícula alfa presente en los materiales y embalaje del propio dispositivo electrónico. Este cambio de estado se puede solucionar a menudo devolviendo el estado del elemento de almacenamiento a su valor original, pero aun siendo poco frecuente, puede afectar a la memoria de configuración y la memoria de datos de las FPGA. A pesar de que los SEU son un fenómeno raro, ocurren con la frecuencia suficiente como para que no puedan ser ignorados, en especial en aplicaciones de alta fiabilidad o alta disponibilidad y se agrava a velocidades de reloj más altas, con lo que los sistemas que requieren de un tiempo de respuesta crítico son más vulnerables a ellos. En la figura 1.1, a la izquierda, se aprecia la reacción del núcleo de silicio al incidir un neutrón sobre él, a la derecha, cómo se ven perturbados los circuitos y se cambia el valor que almacena.

Se pueden construir escudos para proteger las FPGA frente a neutrones con esta cantidad de energía, pero esto no es práctico para la mayoría de los casos, porque la cantidad de material requerido para fabricar este escudo es prohibitiva, siendo de hasta 30 metros de agua para neutrones con energía alta. Por esto los fabricantes de FPGA han desarrollado y puesto a la disposición de los diseñadores una serie de técnicas para proteger y mitigar el efecto de los SEU en sus diseños. Sea cual sea la técnica de protección empleada, surge la necesidad de implementar métodos que nos permitan evaluar la calidad de las soluciones y por tanto la vulnerabilidad de los diseños. Entre estos métodos se encuentran los llamados de inyección de errores, que permiten simular la caída de una partícula cósmica sobre un circuito, y de esta manera poder evaluar las consecuencias de la alteración de la memoria de configuración y/o de datos. Otra manera es someter a las FPGA a

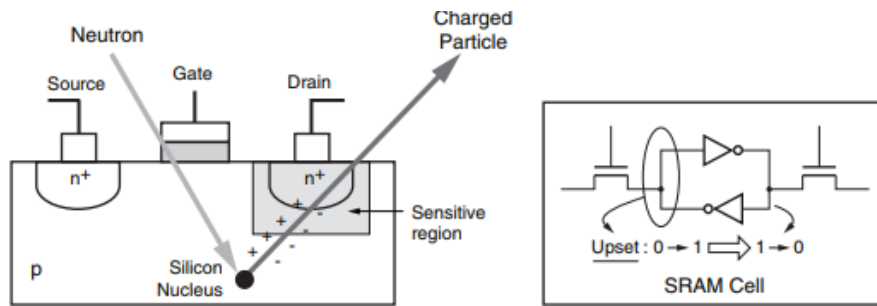


FIGURA 1.1: Comportamiento del sustrato silicio ante la incidencia de un neutrón de alta velocidad. Figura cortesía de [1]

radiación utilizando para ellos aceleradores de partículas. Los aceleradores de partículas pretenden imitar, en cierta manera, la acción de los rayos cósmicos sobre la atmósfera terrestre. Esto se consigue mediante campos electromagnéticos que aceleran partículas cargadas a altas velocidades y las lanzan contra los dispositivos bajo *test*, lo cual produce una lluvia de neutrones sobre ellos. Los aceleradores además proporcionan un entorno mucho más controlado para llevar a cabo los estudios. Con el fin de bombardear con neutrones diversas FPGA para someter a estudio su comportamiento, así como evaluar la calidad de las técnicas de protección empleadas, y de esta manera incrementar la robustez de los diseños que implementan frente a los SEU provocados por la caída de neutrones de alta velocidad, surgió la necesidad de generar y controlar la señal de reloj que gobierna dichos sistemas de una manera externa y a la suficiente distancia para no verse expuesta a los experimentos. Gracias al control absoluto sobre la señal de reloj se podría obtener el comportamiento esperado sobre los dispositivos sometidos a prueba, y comparar estos resultados esperados con los obtenidos tras bombardear las FPGA bajo pruebas en un acelerador de partículas. En la figura 1.2 se puede apreciar la plataforma que se pretende desarrollar y poner bajo estudio, en la que está integrado este Trabajo de Fin de Grado (TFG).

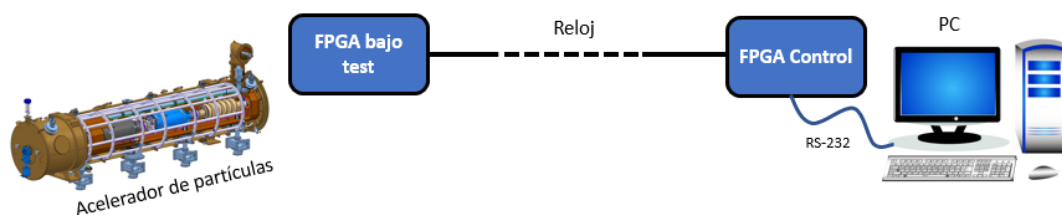


FIGURA 1.2: Diagrama de la plataforma desarrollada.

1.3. Ruido en la transmisión

A la hora de transmitir la señal de reloj a través de líneas de comunicación, ésta queda expuesta al ruido. El ruido consiste en la mezcla de una señal no deseada, con la señal útil que se quiere

transmitir, corrompiendo e incluso haciendo indistinguible dicha señal. Cuando se envían señales digitales por una línea de transmisión, el efecto del ruido se pone de manifiesto en el número de errores que comete el receptor debido a la llegada de una señal completamente distinta a la originalmente transmitida. Cuando además la señal transmitida por el canal gobierna los hitos del sistema receptor, esto puede llegar a tener consecuencias catastróficas en dicho sistema.

En entornos tales como el industrial o el aeroespacial, en plantas nucleares o incluso en los centros donde se pretenden realizar las pruebas con el acelerador de partículas, donde los sistemas embebidos están cobrando un gran protagonismo, existen un sinnúmero de fuentes de ruido. La radiación electromagnética, muy presente en los entornos anteriormente mencionados, y producida en gran parte por la propia alimentación de dichos sistemas, se convierte en una fuente de ruido que afecta a la transmisión de señales eléctricas, como es el caso que se estudiará en este TFG. Cuando la radiación electromagnética incide en un conductor eléctrico, hace que los electrones de su superficie oscilen, generándose de esta forma una corriente alterna cuya frecuencia es la misma que la de la radiación incidente y afectando por tanto a dicha señal.

Además, también en estos entornos se presenta el problema del ruido impulsivo, una de las fuentes de perturbaciones más perjudiciales a la hora de transmitir. Normalmente son picos y oscilaciones de tensión causados por bruscas variaciones de intensidad en el proceso de conexión y desconexión de dispositivos de gran consumo, como puede ser el caso del propio acelerador de partículas, los cuales pueden provocar desde desbalanceo de cargas hasta alteraciones en las diferentes reactancias e inductancias presentes en las líneas de baja potencia utilizadas para la transmisión de señales. El ruido impulsivo se caracteriza porque en señales de amplitud inferior a 5 V de tensión, como es el caso de la transmisión de señales entre FPGA, puede llegar a destruir completamente la posibilidad de comunicación.

Las señales son susceptibles también durante la transmisión al efecto de la atenuación, haciendo que la pérdida de potencia en la transmisión vuelva indistinguible en el circuito receptor la señal recibida.

1.4. Objetivos de este proyecto

Los objetivos específicos de este proyecto, serán, según lo comentado en la sección anterior, los siguientes:

- Análisis de la arquitectura de las FPGA utilizadas.
- Desarrollo de una plataforma *hardware* para el envío fiable de la señal de reloj.
- Estudio de las técnicas de transmisión de señales utilizados para la transmisión fiable de la señal de reloj.
- Toma de resultados de las diferentes técnicas para llegar a transmitir la señal de reloj de manera fiable a cierta distancia.
- Análisis de los resultados obtenidos.

1.5. Fases del trabajo

El proyecto se divide en las siguientes fases, con el fin de cumplir los objetivos arriba mencionados:

1. Se desarrollará el diseño de un circuito que controle la emisión de pulsos de reloj y un circuito receptor que mida la recepción de tales pulsos, ambos embebidos sobre diversas FPGA.
2. Una vez realizado dicho diseño se tomarán resultados para comprobar su eficiencia y posibles limitaciones en la realización de su cometido a diferentes distancias y frecuencias.
3. Se realizarán las modificaciones que implementen las mejoras sobre los diseños originales que permitan transmitir la señal de reloj de una manera fiable con la mayor frecuencia posible a una distancia moderada.
4. Con la nueva implementación se realizarán pruebas del envío de la señal de reloj.
5. Análisis de los datos obtenidos con cada una de las técnicas y modificaciones realizadas.

Capítulo 2

Entorno de desarrollo

En este capítulo se describe el entorno de desarrollo que se ha utilizado para el diseño de las implementaciones *hardware* destinadas al envío, recepción y control de ciclos de reloj entre placas, y que nos servirán para realizar las posteriores modificaciones y la toma de resultados sobre la que se centra este TFG.

Para alcanzar este objetivo se han utilizado diversas FPGA donde se ha implementado tanto el *hardware* del circuito desarrollado como el sistema que controla al mismo, basado en el *softcore* Microblaze, así como el circuito sobre el que se realizarán las posteriores pruebas. Igualmente, se han utilizado herramientas proporcionadas por el fabricante de dichas FPGA, en este caso Xilinx, para el desarrollo del *software* que se ejecuta sobre el sistema de control.

2.1. Entorno *hardware*

Para el desarrollo de este TFG se han utilizado dispositivos dinámicamente reconfigurables, en concreto los modelos de FPGA XC7A35T-1CPG236C de la familia Artix 7, y XC5VLX110T de la familia Virtex 5 para realizar las implementaciones de todos los sistemas de emisión y recepción. La razón de utilizar estas placas es que son las utilizadas por las herramientas de inyección de errores desarrolladas en la Facultad de Informática de la Universidad Complutense de Madrid (Nessy 5.0 y Nessy 7.0).

Dichas FPGA están empotradas en placas de desarrollo, la placa Basys 3 en el caso de la Artix 7, y la placa XUPV5-LX110T en el caso de la Virtex 5, cuyas principales características y arquitecturas se presentarán a lo largo de este capítulo.

2.1.1. Placa de desarrollo Basys 3

La placa Basys 3, la cual se puede observar en la figura 2.1, es una plataforma de desarrollo bastante completa basada en la FPGA Artix 7 de Xilinx del fabricante Diligent®. La tarjeta gira en torno a una FPGA de alto rendimiento, en concreto la XC7A35T-1CPG236C de la serie 7 de Xilinx. Algunas de sus principales características *hardware* son:

- 16 interruptores.
- 16 *leds*.

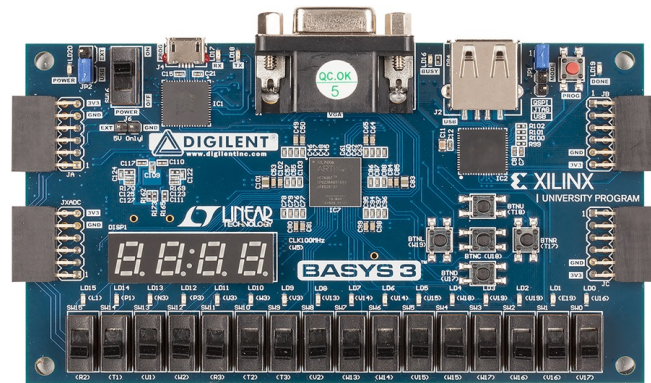


FIGURA 2.1: FPGA Artix 7 empotrada en placa Basys 3.

- 5 botones pulsadores.
- una pantalla de 7 segmentos de 4 dígitos.
- 4 conectores de expansión PMOD.
- 1 puerto USB-JTAG.
- 1 puerto USB-HOST.
- 1 puerto USB-UART.
- 1 puerto VGA.
- 1 memoria FLASH SPI de 32 Mb.

Los conectores PMOD se pueden utilizar para agregar *hardware* adicional, aunque en este caso han sido usados para el envío y recepción de la señal de reloj entre dos placas Basys 3 [2].

2.1.1.1. UART

El protocolo UART (*Universal Asynchronous Receiver-Transmitter*) es un protocolo de comunicación muy usado para controlar los puertos y dispositivos serie. Este transforma los datos del sistema a formato serie, toma estos *bytes* de datos y posteriormente transmite los *bits* individualmente de forma secuencial, reduciendo el coste de implementación con respecto a la transmisión en paralelo, mediante utilización de tan solo dos cables, uno para envío y otro para recepción.

Una de las características de la tarjeta Basys 3 es que cuenta con un puerto UART gracias a que incorpora un *chip* FTDI FT2232HQ conectado al puerto microUSB. Dicho circuito integrado permite la comunicación de la tarjeta Basys 3, mediante transmisión estándar RS-232, con una computadora. El FT2232HQ está conectado a la FPGA por medio de los pines TxD y RxD. Las conexiones entre el FT2232HQ y la FPGA Artix-7 se muestran en la figura 2.2.

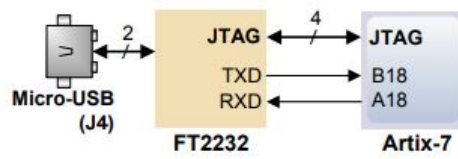


FIGURA 2.2: Conexión entre el *chip* FT2232HQ y Artix 7. Figura cortesía de [2].

2.1.1.2. JTAG

JTAG es un acrónimo para *Joint Test Action Group*, siendo este el nombre común utilizado para la norma IEEE 1149.1 titulada *Standard Test Access Port and Boundary-Scan Architecture*. Fue diseñado originalmente para circuitos impresos, y utilizado para la prueba y configuración de submódulos de circuitos integrados, siendo muy útil también como mecanismo para depuración de aplicaciones en sistemas embebidos.

Una interfaz JTAG es una interfaz especial de cuatro o cinco pines agregados a un *chip*. Está diseñada de tal manera que varios *chips* pertenecientes a la misma tarjeta puedan tener sus líneas JTAG conectadas en *daisy chain*, como se muestra en la figura 2.3. Así una sonda de prueba JTAG mediante la conexión a un solo puerto JTAG puede acceder a todos los *chips* en un mismo PCB (*Printed Circuit Board*). Durante la fase de programación mediante JTAG el archivo *bitstream* se transfiere desde una computadora a la FPGA.

El *chip* FT2232HQ, incorporado en la placa Basys 3, se utiliza tanto como controlador del puerto USB-JTAG como para llevar a cabo la configuración de la FPGA. Aun siendo el mismo puerto físicamente, no ofrece problema alguno, ya que estos dos sistemas de comunicación funcionan bien de forma simultánea, lo que significa que si en el diseño se requiere de la comunicación con una computadora por medio de puerto UART, el puerto USB-JTAG no interferirá en su ejecución ni viceversa. El puerto *microUSB* también se puede utilizar para alimentar la placa Basys 3.

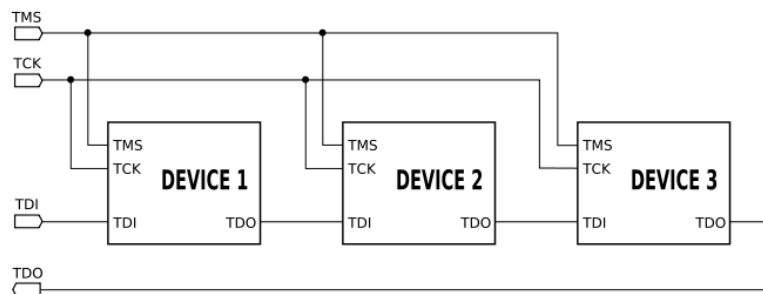


FIGURA 2.3: Interfaz JTAG conectada en *daisy-chain*.

2.1.1.3. Conectores PMOD de la placa Basys 3

La Basys 3 cuenta con 4 conectores PMOD que ofrecen la posibilidad de añadir *hardware* a la tarjeta. Cada conector ofrece 8 señales de datos, dos conexiones a tierra y dos a 3.3v. El puerto de expansión PMOD, etiquetado como JXADC está conectado a los pines de entrada analógica auxiliar de la FPGA. Dependiendo de la configuración, este conector se puede utilizar para ingresar señales analógicas diferenciales al conversor analógico digital dentro de la Artix-7 (XADC). Cualquiera o todos los pares en este conector pueden configurarse como entrada analógica o entrada-salida digital.

Los ocho pares de cada conector se dirigen a través del circuito impreso de la placa, acoplados estrechamente, hasta conectarse con los correspondientes pines de la FPGA para conseguir una mayor inmunidad contra el ruido analógico. Además, cada par tiene un filtro *antialiasing*, que se distribuye en el PCB (*Printed Circuit Board*) para limpiar las señal antes de hacer una conversión analógica-digital.

2.1.2. Placa de desarrollo XUPV5-LX110T

La placa de desarrollo XUPV5-LX110T gira en torno a una FPGA de la familia Virtex 5, la XC5VLX110T como ya se ha mencionado y que se puede observar en la figura 2.4.

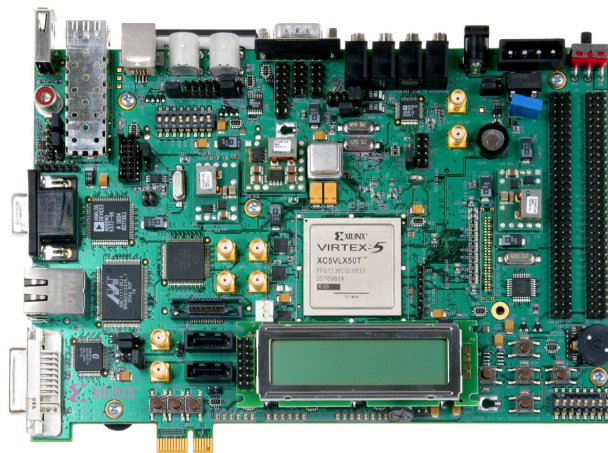


FIGURA 2.4: FPGA Virtex 5 empotrada en placa XUPV5-LX110T.

La XC5VLX110T es una placa de desarrollo y evaluación de ámbito general y entre las características *hardware* que presenta se encuentran:

- 16 interruptores divididos en dos bancos.
- 13 leds *leds*.
- 5 botones pulsadores.
- 1 puerto JTAG.
- 1 puerto USB-HOST.

- 1 puerto RS232.
- 1 puerto PS/2.
- 8 conectores SMA.
- 1 memoria FLASH SPI de 32Mb(4MB).

De los ocho conectores SMA, cuatro están destinados para el envío y recepción de relojes diferenciales. Cuatro de estos conectores permiten enviar y recibir, desde y hacia la FPGA, señales de reloj diferenciales a través de ellos[7] y poseen una impedancia de 50 Ω . La Virtex 5 puede configurarse con una impedancia de terminación de 100 Ω como se verá en capítulos posteriores. Esto permite que la FPGA pueda enviar un reloj de precisión a un dispositivo externo. En la figura 2.5 se puede apreciar la conexión entre la FPGA Virtex 5 y el circuito integrado destinado al control de los puertos SMA por donde se emite el reloj diferencial. También se puede apreciar la presencia de una resistencia de terminación de 100 Ω .

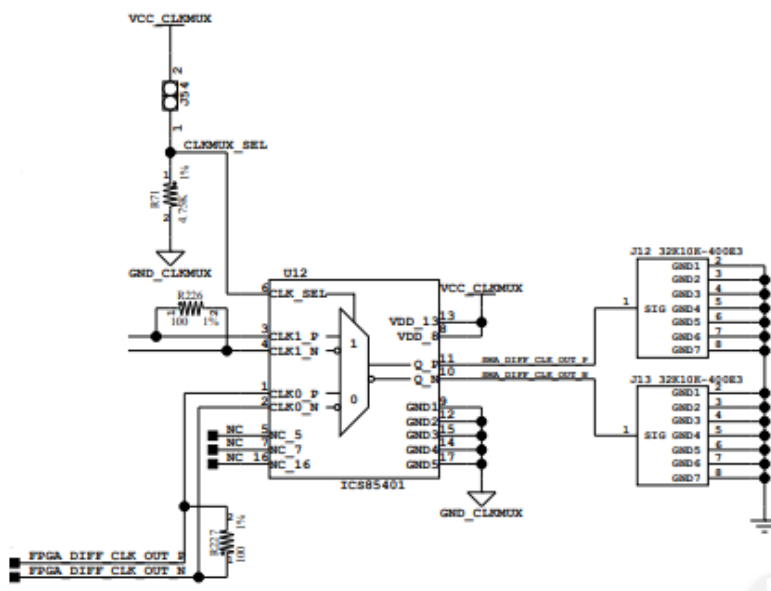


FIGURA 2.5: Circuito integrado destinado al control de los conectores SMA de salida de reloj. Figura cortesía de [3].

2.1.3. FPGA

Como se ha indicado en los anteriores apartados este proyecto gira en torno a la perdida de ciclos de reloj al ser enviado entre FPGA. Las FPGA son dispositivos electrónicos reconfigurables constituidos por CLB (*Configurable Logic Block*) en su mayoría, cuya funcionalidad e interconexión puede adaptarse múltiples veces mediante el volcado de un mapa de *bits* del diseño en una memoria. Los bloques individuales están constituidos por elementos que les permiten adoptar distintas funciones de transferencia.

Una de las mayores ventajas que ofrecen las FPGA, cuya arquitectura básica se puede observar en la figura 2.6, frente a los convencionales circuitos ASIC, es que para diseñar un sistema no es necesario su fabricación, sino tan solo el volcado del mapa de *bits* en la FPGA; además los costes de desarrollo y adquisición son mucho menores.

La capacidad de reprogramación, flexibilidad y capacidad de proceso paralelo, convierte a las FPGA en dispositivos electrónicos muy atractivos en múltiples sectores, siendo dispositivos que se emplean habitualmente en sistemas aeroespaciales, sistemas de procesamiento de señales, aplicaciones que requieren tiempos de respuesta críticos, dispositivos sanitarios y dispositivos de control de maquinaria industrial, entre otros.

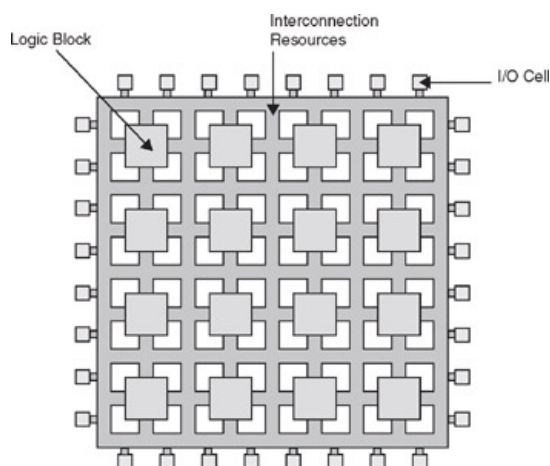


FIGURA 2.6: Arquitectura básica de una FPGA.

2.1.3.1. Arquitectura Xilinx Artix 7

La familia Artix 7 presenta la mitad de consumo en comparación con la anterior generación de FPGA de Xilinx, sin dejar de ofrecer una funcionalidad avanzada para aplicaciones de alto rendimiento. Proporciona uno de los mejores ratio de rendimiento por vatio en su gama, siendo por ello muy utilizada en equipos médicos portátiles, radios militares e infraestructuras inalámbricas.

Para la implementación de los diferentes diseños desarrollados en este trabajo se ha usado la FPGA XC7A35T-1CPG236C de la familia Artix 7. A continuación se describen los fundamentos de su arquitectura:

- **CLB (Configurable Logic Blocks):** son los elementos de las FPGA que están destinados a implementar la lógica del diseño. Los CLB están formados por puertas lógicas, multiplexores, LUT y Flip-Flops. En la familia Artix 7 cada CLB está constituido por dos *slices*, cada uno de los cuales está formado a su vez por cuatro LUT (*Look-Up Table*), ocho registros de los cuales cuatro pueden ser configurados como latches y los multiplexores correspondientes. Estos *slice* son denominados SLICEL. Existen otros *slices* denominados SLICEM, los cuales presentan dos funcionalidades adicionales: almacenar datos usando RAM (*Random Access Memory*) distribuida, y el desplazamiento de datos mediante el uso de registros de 32

bits. Cada CLB puede estar formado por dos SLICEL o por un SLICEL y un SLICEM. En la figura 2.7 se puede observar el diseño y componentes de un SLICEL de la familia Artix 7.

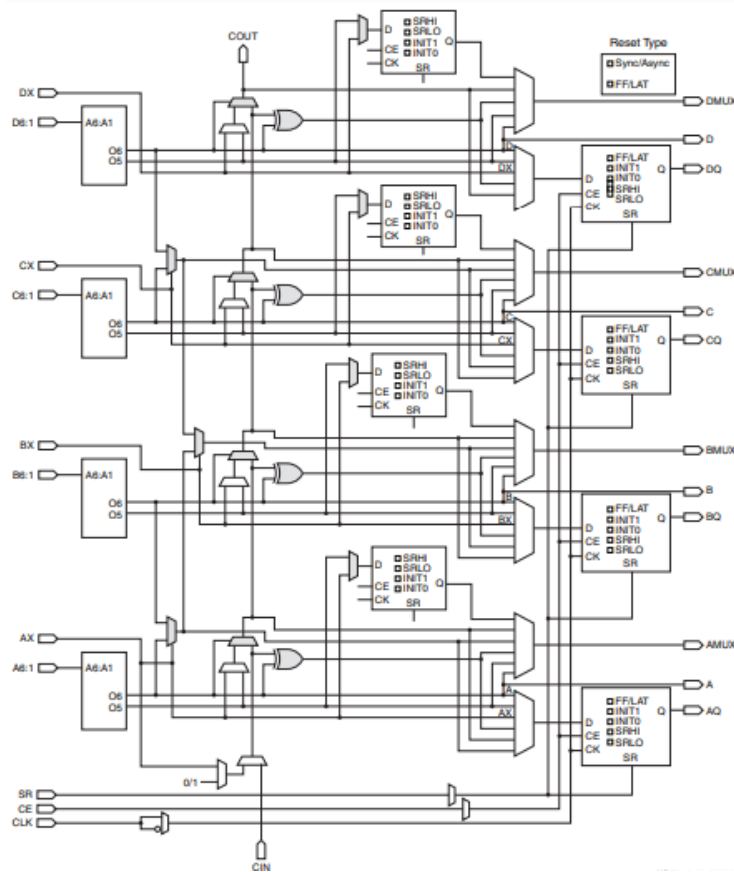


FIGURA 2.7: SLICEL perteneciente a un CLB de la familia Artix 7. Figura cortesía de [4].

Las LUT están compuestas por celdas de memoria y multiplexores. Básicamente son tablas de almacenamiento que ofrecen una salida en función de la entrada, implementando de esta manera la función lógica que se les haya programado. Las LUT se utilizan por tanto para implementar funciones combinacionales. En la familia Artix 7 las LUT se pueden configurar como LUT de seis entradas con una salida o como dos LUT de cinco entradas con salidas separadas.

- **PSM (Programmable Switch Matrix):** es la red de interconexión reconfigurable que se encarga de comunicar los distintos elementos de la FPGA mediante canales horizontales y verticales distribuidos en ella.
- **CB (Connection Block):** son los bloques encargados de conectar los CLB a la PSM.
- **IOB (Input Output Blocks):** son bloques que sirven de unión entre los bloques lógicos internos y los pines terminales de entrada/salida. Cada IOB de las FPGA de la serie 7 tiene 4 pines CCIO (Clock Capable Input Output) [5], que se identifican por contener CC en su

identificador. En la figura 2.8 se pueden apreciar los IOB de la FPGA Artix 7 con empaquetamiento CPG236.

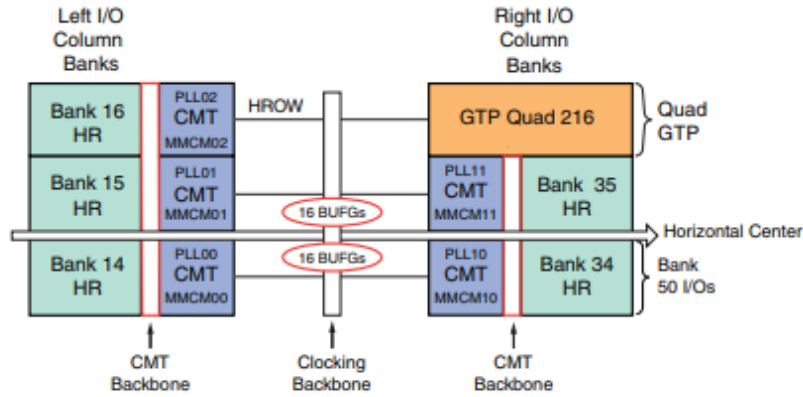


FIGURA 2.8: Distribución de los IOB en la FPGA XC7A35T 1CPG236C. Figura cortesía de [5].

- **CMT(Clock Management Tiles):** para manipular la señal de reloj, las FPGA de la serie 7 cuentan con hasta 24 CMT, cada uno compuesto por un MMCM (*Mixed-Mode Clock Manager*) y un PLL (*Phase Locked Loop*) como se aprecia en la figura 2.9. Estos módulos en colaboración con los *buffers* de distribución de reloj, permiten que dicha señal llegue de manera inalterada y simultánea a los diferentes componentes de las FPGA que requieren de ella. Los módulos MMCM y PLL sirven tanto para sintetizar una amplia gama de frecuencias, como para generar señales de reloj con fase desplazada, y llevar a cabo la función de filtrado de efecto *jitter* para los relojes procedentes de fuentes externas o internas y la eliminación del efecto *skew* de las señales de reloj [6]. En el capítulo 4 se hablará más en profundidad sobre estos efectos adversos.

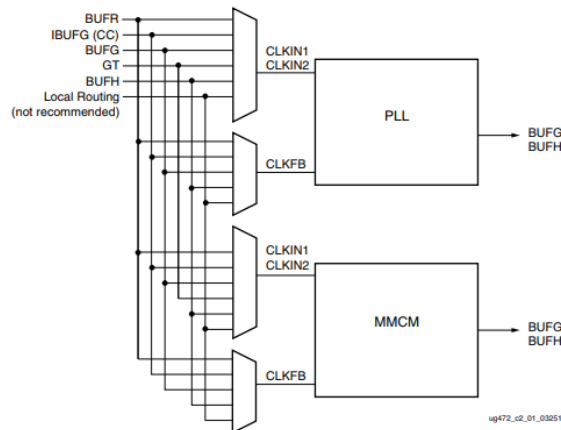


FIGURA 2.9: Diagrama de bloques de un CMT de la serie 7 de Xilinx. Figura cortesía de [6].

2.1.3.2. Arquitectura Xilinx Virtex 5

Debido a que parte del estudio de este TFG se ha llevado a cabo sobre una FPGA Virtex 5 XC5VLX110T, a continuación, se nombrarán las características y componentes principales de su arquitectura que se han tenido en cuenta:

- **CLB:** la XC5VLX110T presenta una matriz de 160 x 54 CLB totales, con un máximo de 1.120 Kb de memoria RAM entre todos ellos. Además, estos CLB están divididos en dos SLICE, que están conectados a los demás elementos de la FPGA a través de una matriz de interconexión. Estos SLICE son también de tipo SLICEL y SLICEM como pasa en la Artix 7. Cada CLB tiene a su vez un total de las 4 LUT, cada una de las cuales internamente ha sido extendida a 6 entradas, pudiendo ser configuradas como funciones booleanas de 6 entradas y 1 salida o como 2 funciones booleanas de 5 entradas y 1 salida para cada función.
- **DSP (*Digital Signal Processing*):** Los *slice* DSP son módulos cuya función es realizar operaciones aritmético lógicas típicas de circuitos para procesamiento de señal digital. Están compuestos por un módulo multiplicador, un sumador de 48 *bits* y un acumulador.
- **PSM:** de la misma forma que pasaba en la Artix 7, proporciona una red de interconexión reconfigurable que conecta los diferentes bloques de la FPGA.
- **IOB:** son los bloques que comunican la FPGA con el exterior y se disponen rodeando la matriz de CLB. Se pueden configurar para que se comporten como puertos de entrada/salida.
- **CMT:** para manipular la señal de reloj, la Virtex 5 cuenta con 6 CMT compuestos cada uno por 2 DCM y 1 PLL, a diferencia de la Artix 7. Los DCM son módulos que permiten la manipulación de la señal de reloj, pudiendo modificar su frecuencia multiplicándola o dividiéndola, reacondicionar el *duty cycle* y eliminar los efectos del *clock skew*. El PLL es un módulo con realimentación cuya función es que la señal de reloj generada sea lo más perfecta posible. Estos módulos junto los *buffers* de distribución de reloj repartidos por la FPGA aseguran la correcta sincronización de los elementos involucrados en los diseños. En la figura 2.10 se puede apreciar un CMT de la familia Virtex 5 de Xilinx.

2.1.3.3. Arquitectura de reloj en las FPGA de Xilinx

En las FPGA el recurso de reloj suministrado debe de cumplir una serie de características que aseguren el buen comportamiento de las implementaciones diseñadas, a destacar:

- Bajo *skew*.
- Baja fluctuación (*jitter*).
- Baja latencia.
- Preservación del ciclo de trabajo.

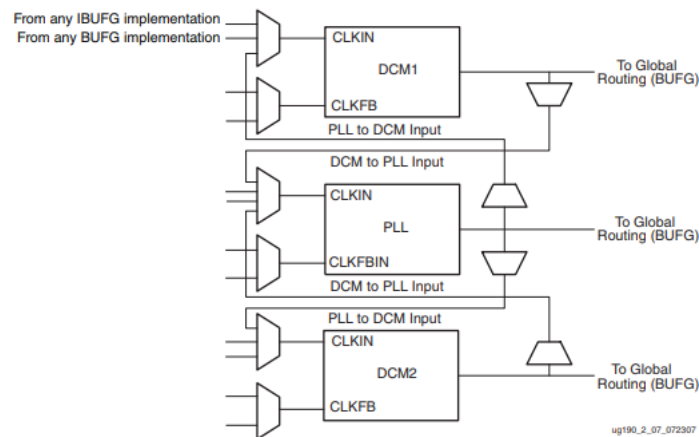


FIGURA 2.10: Diagrama de bloques de un CMT de la familia Virtex 5 de Xilinx. Figura cortesía de [7].

Para conseguir estos objetivos, las FPGA de Xilinx, distribuyen la señal de reloj en su interior sirviéndose de una serie de *buffers* y módulos que aseguren las características anteriormente mencionadas. Como se aprecia en la figura 2.11, la FPGA queda dividida en regiones de reloj.

Las FPGA de la serie 7 tienen entre 2 y 24 regiones de reloj y las de la familia Virtex 5 entre 8 y 12. Atravesando la FPGA verticalmente y separando las regiones de reloj este y oeste se encuentra la columna global de distribución de reloj conectada a las HROW mediante los BUFG. Cada región de reloj abarca un total de 50 IOB y 50 CLB de altura, y la mitad del *chip* de anchura. En cada región de reloj se encuentra un CMT y tiene una fila horizontal de reloj HROW atravesándola, la cual gracias a los recursos de distribución de reloj, *buffers* que se encuentran dentro de dicha región, permite distribuir la señal a los diversos componentes que la constituyen, como se ve en la figura 2.12.

De una manera parecida, cada IOB contiene pines de entrada con capacidad de reloj para llevar los relojes del usuario a los recursos de distribución del reloj, es decir, los **buffers**. Dichos pines se distinguen por contener CC en su identificador, como ya se ha dicho anteriormente. Junto con los *buffers* de reloj dedicados, la entrada con capacidad de reloj lleva los relojes del usuario a:

- Líneas de reloj globales en la misma mitad superior o inferior del dispositivo.
- Líneas de reloj de E/S dentro del mismo banco de E/S y bancos de E/S adyacentes verticalmente.
- Líneas de reloj regionales dentro de la misma región de reloj y regiones de reloj adyacentes verticalmente.
- CMT dentro de la misma región de reloj.

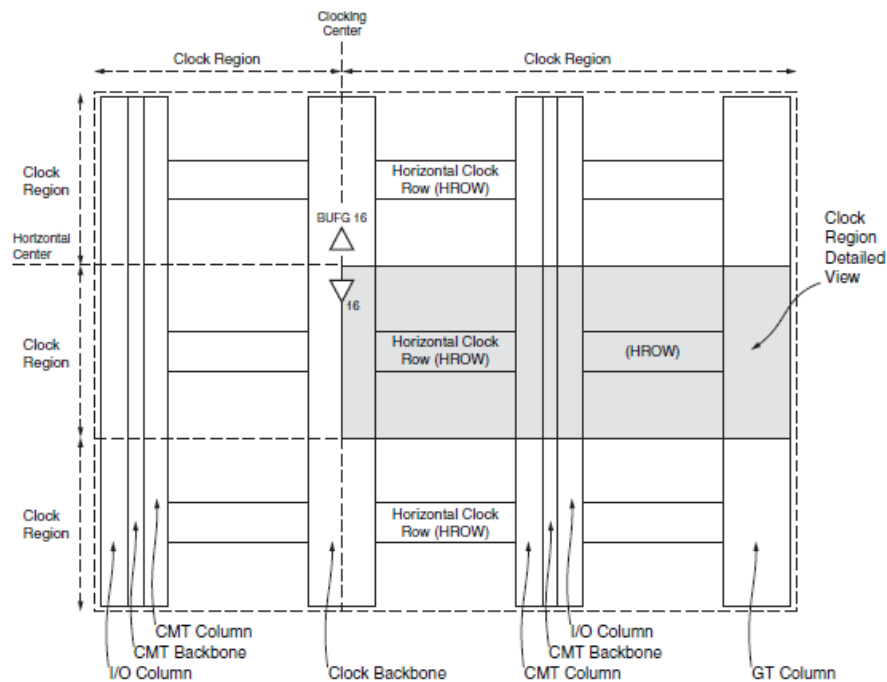


FIGURA 2.11: Arquitectura de reloj de la serie 7 de Xilinx. Figura cortesía de [6].

2.2. Herramientas *software* para el desarrollo

Para la realización de la parte relacionada con la implementación sobre la FPGA y el desarrollo de la parte *software* de la plataforma de control y emisión de pulsos de reloj, se han utilizado distintas herramientas de Xilinx, entre ellas Vivado Design Suite 17.4, Xilinx EDK 14.7, Xilinx SDK 14.7 e Impact 14.7. En las siguientes subsecciones se indica el uso prestado por cada una de estas herramientas para el desarrollo de este proyecto.

2.2.1. Vivado Design Suite

Vivado *Design Suite* es un juego de herramientas producido por Xilinx para la síntesis y análisis de diseños HDL (*Hardware Description Language*), reemplazando a Xilinx ISE y proporcionando una serie de características adicionales para el desarrollo de sistemas SoC. Constituye una reescritura básica y una nueva concepción del flujo de diseño completo, en comparación con ISE [15]. Este *software* fue desarrollado para FPGA de mayor capacidad, e incluye la funcionalidad de síntesis de alto nivel que permite a los desarrolladores compilar coprocesadores con una descripción basada en C. Es importante resaltar que la FPGA XC7A35T no es compatible con EDK y por tanto debe configurarse con Vivado.

Vivado permite a los desarrolladores sintetizar sus diseños, analizar tiempos, examinar los diagramas RTL (*Register-Transfer Level*), realizar simulaciones y configurar los dispositivos destino. Vivado está estrechamente vinculado a la arquitectura de las FPGA de Xilinx, con lo que no se puede usar con productos FPGA de otros proveedores. Es compatible con las familias UltraScale, Virtex 7, Kintex 7, Artix 7, y Zynq 7000 de Xilinx, por lo que para configurar la Virtex 5 ha sido

2.2.4. Impact

Como se explicará posteriormente, debido a las limitaciones que surgirán en la optimización, será necesaria la utilización de Impact para llevar a cabo la programación de la FPGA Virtex5 empotrada en la placa de desarrollo XUPV5-LX110T. Impact es un *software* desarrollado por Xilinx y que es utilizado para programar y configurar las FPGA que no son compatibles con Vivado. Permite la comunicación y configuración de las FPGA de varias formas desde una computadora, mediante *boundary-scan*, de una manera bastante sencilla. Una vez que se ha establecido la conexión entre la FPGA e Impact, solo hay que seleccionar el *bitstream* deseado y la FPGA se programará con éste.

Capítulo 3

Hardware para la transmisión fiable de una señal de reloj entre FPGA

En este capítulo se entra en detalle en el diseño de la plataforma que se ha creado para conseguir transmitir de manera fiable la señal de reloj entre FPGA. Como se adelantó en el capítulo 2, se han desarrollado diferentes prototipos los cuales implementan diferentes técnicas para proteger la señal frente al ruido electromagnético. En la figura 3.1 se puede observar, la arquitectura de la plataforma emisora desarrollada se basa en un sistema *hardware* basado en un microprocesador implementado en una FPGA, cuyo diseño se describe a lo largo de este capítulo.

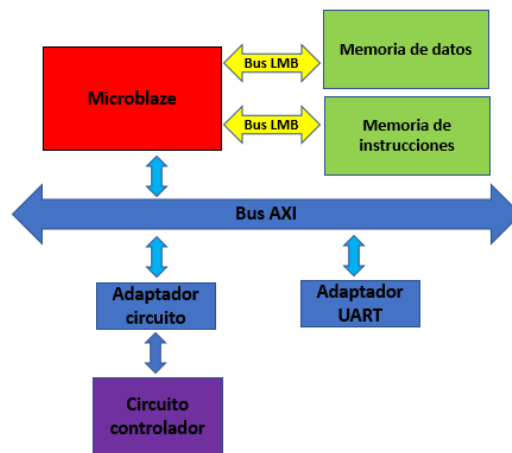


FIGURA 3.1: Arquitectura *hardware* de la plataforma que controla la emisión de ciclos de reloj.

El procesador del sistema *hardware* emisor está implementado con MicroBlaze y su componente *software* se ha programado con el lenguaje de alto nivel C. También se ha desarrollado un interfaz para comunicarse con un PC. Para programar las FPGA con los diseños realizados en el lenguaje de especificación *hardware* VHDL se ha utilizado el puerto micro USB-JTAG de la placa Basys 3 y el JTAG de la placa XUPV5-LX110T respectivamente, mientras que para la comunicación de datos se utilizara la UART de las placas empleando para ello el protocolo RS-232.

3.1. Arquitectura *Hardware*

Como ya se ha mencionado, se ha utilizado Vivado para el diseño de los distintos sistemas *hardware*. Fundamentalmente, el sistema que controla la emisión de la señal de reloj está formado por un procesador MicroBlaze conectado mediante un bus LMB (Local Memory Bus) a sus memorias de instrucciones y datos. Además, cuenta con un módulo que gestiona la comunicación con la UART de la placa utilizada y una interfaz AXI (*Advanced Extensible Interface*) para la comunicación con el prototipo desarrollado. Todos estos módulos están conectados a un bus AXI por el que se comunicarán a una frecuencia de reloj que variará durante la realización de las diversas pruebas y toma de datos.

3.1.1. MicroBlaze

MicroBlaze es un *softcore* diseñado por Xilinx para sus FPGA. MicroBlaze está descrito en HDL y se implementa en las FPGA de Xilinx. MicroBlaze es similar a la arquitectura DLX basada en arquitectura RISC (*Reduced Instruction Set Computer*), pudiendo haber varias instrucciones simultáneamente en el *pipeline* en distintas etapas, lo que permite en condiciones ideales, sacar una instrucción por ciclo de éste. Es importante destacar la gran capacidad de MicroBlaze para ser configurados por el programador: la profundidad del *pipeline*, los periféricos que integra, la unidad de que administra memoria y las interfaces de bus. Posee registros de propósito general de 32 *bits*, instrucciones de 32 *bits* con 3 operandos y 2 modos de direccionamiento [8].

MicroBlaze posee un sistema de interconexión versátil para admitir una variedad de aplicaciones integradas. El bus de entrada-salida principal de MicroBlaze, el bus PLB, es un bus mapeado en la memoria del sistema con capacidad maestro/esclavo. Las versiones más recientes de MicroBlaze admiten la especificación AXI.

Para este proyecto se ha usado la versión 8.5.b de MicroBlaze para el proceso y gestión del *hardware* desarrollado. Este procesador está escuchando el puerto serie a la espera de algún comando enviado desde el sistema *software* del PC y según sea el comando recibido se encargará de ejecutar el programa situado en una memoria RAM interna de la FPGA a la que está conectada por el bus LMB (*Local Memory Bus*), que, al igual que AXI, también será descrito en las siguientes subsecciones.

3.1.2. Bus AXI

En 2003, ARM introdujo AXI, como parte de la tercera generación de AMBA (*Advanced Microcontroller Bus Architecture*), para alcanzar un mayor rendimiento en las interconexiones. En 2010, se lanzó AXI4, la segunda versión de AXI, la cual se ha usado para el desarrollo de este proyecto [9]. El protocolo AXI está dirigido a diseños de sistemas con alta frecuencia de reloj y alto rendimiento e incluye características que lo hacen adecuado para la interconexión submicrométrica de alta velocidad.

El ARM AMBA es una especificación de interconexión abierta para la conexión y gestión de bloques funcionales en diseños de SoC. Facilita el desarrollo de diseños multiprocesador con un gran número de controladores y periféricos. Desde su inicio, el alcance de AMBA, a pesar de su

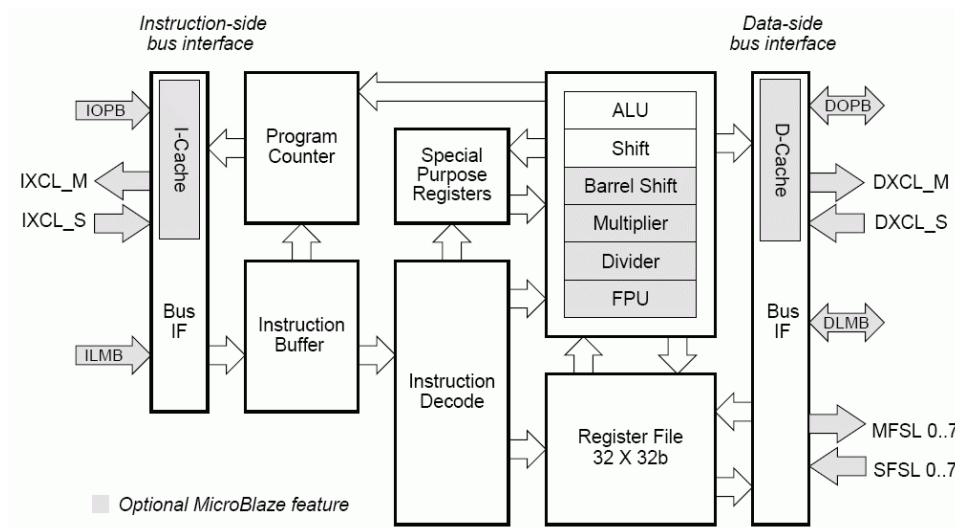


FIGURA 3.2: Arquitectura MicroBlaze. Figura cortesía de [8].

nombre, ha ido más allá de los dispositivos de microcontroladores. En la actualidad, AMBA se usa ampliamente en una gran variedad de dispositivos ASIC y SoC.

Hay tres tipos de interfaces AXI4:

- AXI4: para requisitos de alto rendimiento.
- AXI4-Stream: para transmisión de datos de alta velocidad.
- AXI4-Lite: para comunicación mapeada en memoria de bajo rendimiento, por ejemplo, la que se hace desde registros de control y estado. A diferencia de la interfaz AXI4 no soporta modo ráfaga. Esta interfaz será la que se utilizará a lo largo del desarrollo de este proyecto.

Los maestros y esclavos AXI mapeados en memoria se pueden conectar juntos usando un bloque de interconexión. La comunicación se da entre un maestro con un esclavo. La interfaz AXI4-Lite consta de cinco canales diferenciados para las siguientes funciones: lectura de direcciones, escritura de direcciones, lectura de datos, escritura de datos, escritura de respuesta. La transferencia de datos puede realizarse en ambas direcciones, entre el maestro y el esclavo, de manera simultánea, siendo este tamaño de datos variable. AXI4-Lite solo permite una transferencia de datos por transacción. Añadiendo el módulo AXI *Interconnect* se consigue conectar a MicroBlaze uno o más dispositivos maestros mapeados en memoria, a los cuales podemos conectar uno o más dispositivos esclavos también mapeados en memoria, como será para el caso de este proyecto los módulos UARTlite, que gestionará la comunicación con el exterior a través del puerto serie, y el propio periférico que se ha diseñado específicamente.

3.1.3. Bus LMB

LMB (*Local Memory Bus*) es un bus especial usado para conectar MicroBlaze con dispositivos de alta velocidad, en el caso de este proyecto con la memoria BRAM. Se han usado dos buses LMB

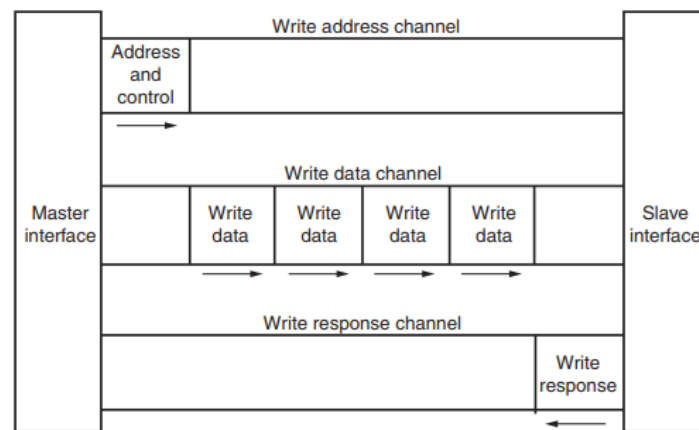


FIGURA 3.3: Transacción de escritura AXI. Figura cortesía de [9].

conectados al MicroBlaze que van a dos controladores diferenciados de memoria BRAM, uno de datos y otro de instrucciones, para poder trabajar con ambos simultáneamente.

3.1.4. Adaptador UART AXI

A través de este módulo se realiza la comunicación de la placa con el exterior mediante el protocolo de comunicación RS-232. Mediante su adición se proporciona un interfaz de comunicación con el bus AXI y para el control de la transferencia de datos serie asíncrona. De esta manera se consigue que MicroBlaze puede intercambiar datos y comandos con un computador.

3.1.5. Asistente de sincronización

El *Clocking Wizard* es un *core* desarrollado por Xilinx que se puede generar e incluir en los diseños utilizando Vivado y EDK. Permite la creación de una señal de reloj a partir de otra señal de entrada, en este caso el reloj generado por el oscilador de cuarzo presente en la FPGA. Para distribuir de manera correcta esta señal por la red de reloj asegurando así una frecuencia, fase y un ciclo de trabajo con *jitter* reducido, se usan las primitivas MMCM y PLL repartidas por el diseño RTL. Además, este módulo permite la creación de diferentes señales de reloj y facilita la gestión de las siguientes características [10] por parte de los desarrolladores:

- Síntesis de frecuencias: permite que los relojes de salida del módulo tengan diferentes frecuencias que el reloj de entrada.
- Alineación de fase: para las diferentes primitivas de reloj asociadas al módulo.
- Minimiza la potencia: permite minimizar la cantidad de energía necesaria para la primitiva de reloj a expensas de la frecuencia, fase o la precisión del ciclo de trabajo.
- Cambio de fase dinámico: permite cambiar la relación de fase en los relojes de salida.

- Reconfiguración dinámica: permite cambiar la programación de la primitiva de reloj después de la configuración del dispositivo. Cuando se elige esta opción, la interfaz AXI4-Lite se selecciona por defecto para reconfigurar el reloj.
- Equilibrado: permite la optimización del *jitter*.
- Minimización del *jitter* de entrada y salida: minimiza la inestabilidad en los relojes de salida, a costa de la potencia y, posiblemente, del error de fase del reloj de salida generado.
- Arranque y secuenciamiento seguro del reloj: útil para obtener un reloj estable y válido a la salida del módulo. También permite habilitar relojes en un orden de secuencia particular.
- Monitorización: de hasta 4 señales de reloj.

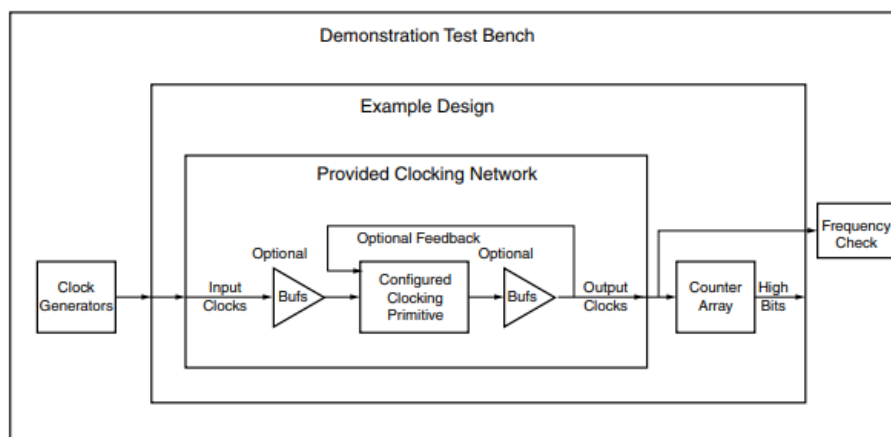


FIGURA 3.4: Diagrama de bloques del asistente de sincronización. Figura cortesía de [10].

En la figura 3.4 se puede apreciar el diagrama de bloque en el que se fundamenta el funcionamiento del asistente de sincronización.

3.1.6. Circuito controlador de ciclos de reloj

Este circuito basa su comportamiento en una máquina de estados finitos de tipo Moore, cuyo diseño aparece en la figura 3.5. Esta máquina de estados se encarga de controlar la emisión de la señal de reloj desde el circuito destinado al control. Para ello se comunicará con MicroBlaze a través de tres registros de 32 *bits*. En dos de estos registros se escribe vía *software* el número total de ciclos a enviar, pudiendo de esta manera enviar hasta un total de 2^{64} ciclos de reloj. Este dato se introduce a través de un terminal desde un PC, que se comunica con la FPGA a través del puerto serie. Por otra parte, el registro restante se utiliza como registro de control y en él se escribe la señal de enable desde *software*. Para una correcta sincronización entre el *software* y el *hardware*, puesto que el *hardware* suele ser bastante más rápido que el *software*, es necesario instanciar y diseñar un módulo detector de flancos para, de esta manera, poder detectar flancos cuando la señal

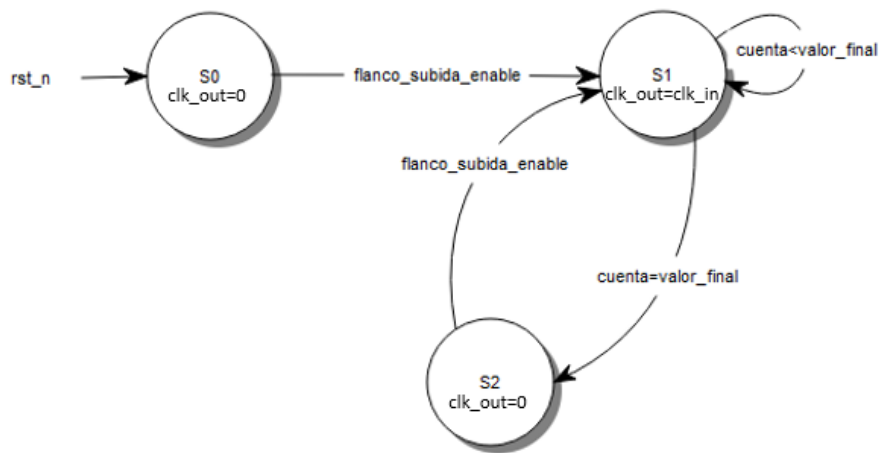


FIGURA 3.5: Máquina de estados que controla la emisión de ciclos de reloj.

de *enable* cambia de 0 a 1, siendo esta señal la que gobierna el cambio de estados de la máquina Moore.

Como se verá posteriormente en el capítulo 4, mediante la instanciación del componente OBUFDS, en este circuito emisor de ciclos de reloj es posible emitir el reloj de salida del circuito de manera diferencial, lo cual será clave para poder transmitir la señal de manera fiable.

3.1.7. Circuito bajo prueba

Para poder hacer las posteriores pruebas y toma de resultados se ha diseñado un circuito receptor que será básicamente un contador de 64 *bits*. Con este diseño se programa otra placa basada en FPGA, de las utilizadas en el trabajo, volcando el archivo *bitstream* generado a través de JTAG. Dicho contador se incrementa en uno cada vez que recibe un ciclo de reloj procedente de la otra placa. También será necesario una instancia del componente IBUFDS que se utiliza para construir la señal de reloj a contar a partir de la señal diferencial que se recibe.

3.1.8. Sistema Software

El sistema *software* consiste en una sencilla interfaz que permite comunicar la FPGA con un PC a través de un terminal, para poder indicar desde éste el número de ciclos que serán enviados desde una FPGA a la otra. Dicho *software* lee de un terminal datos y comandos como por ejemplo el número de ciclos a enviar y, una vez obtenidos estos, escribe en los registros del circuito emisor esta cantidad; una vez hecho esto, escribe en otro registro de control el valor de la señal de *enable* para indicar al *hardware* que ya pueden ser enviados dichos ciclos de reloj desde el circuito emisor.

Capítulo 4

Metodología

4.1. Introducción a la metodología

En este capítulo se muestran las técnicas y metodología que se han utilizado para realizar optimizaciones sobre los diferentes prototipos que han sido desarrollados durante este TFG. Dichas optimizaciones tienen base en una serie de problemas que surgen a la hora de transmitir el reloj entre placas basadas en FPGA de una manera fiable. Estos problemas están asociados a la aparición de ruido en la señal. Para abordar el objetivo del TFG, se realizará, en primer lugar, un estudio de la cantidad de ciclos de reloj perdidos, los cuales son emitidos desde una placa de desarrollo emisora que implementa el diseño del circuito destinado al control de la emisión de ciclos de reloj, hacia una placa receptora, y se tomarán las muestras pertinentes utilizando diferentes longitudes de cables que comunican ambas placas de desarrollo.

En una segunda fase se implementarán mejoras sobre los diseños, tanto de los circuitos emisores como receptores, así como de los elementos de interconexión pasivos, para evitar la pérdida de ciclos, y conseguir transmitir la señal de manera fiable mediante las diferentes técnicas estudiadas en este capítulo.

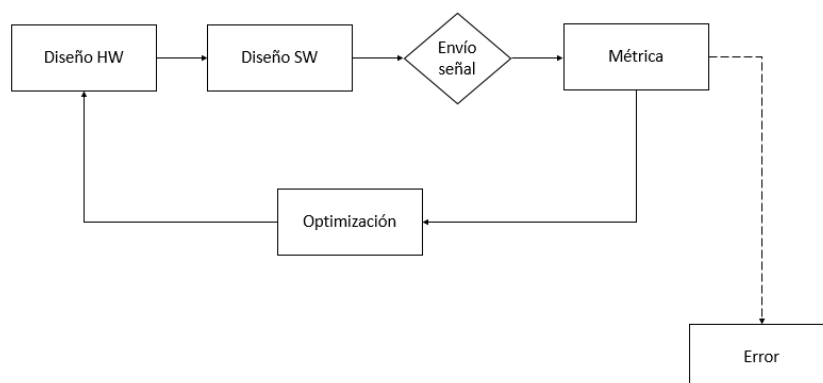


FIGURA 4.1: Diagrama que muestra la metodología de trabajo para realizar las mejoras y mediciones.

En la figura 4.1 se muestra el esquema de optimización mono-objetivo que se seguirá para llevar a cabo las consecutivas transformaciones sobre el diseño sobre el que se realizarán las mediciones de resultados. Se realizará una optimización en bucle hasta que se consigan unos resultados que garanticen la transmisión fiable de la señal de reloj.

La métrica será el número de pulsos perdidos entre los circuitos receptor y emisor cuyos diseños son implementados en las placas basadas en FPGA. Tras la toma de resultados se procederá a realizar las diferentes optimizaciones sobre el circuito y los consecuentes cambios en los diseños *hardware*, sobre los que se volverá a medir la cantidad de ciclos de reloj perdidos en la transmisión.

4.2. Optimización de los elementos de transmisión pasivos.

Tras la medición de los resultados del primer prototipo desarrollado, el cual no implementaba ninguna técnica para minimizar la pérdida de la señal de reloj tanto en el circuito emisor como en el receptor, se procedió a mejorar los elementos pasivos que formaban parte del experimento, siendo el principal elemento pasivo en este caso el cable que se utiliza para comunicar los circuitos emisor y receptor. Para ello se procedió a seleccionar el cable que comunica ambos dispositivos para paliar los efectos adversos e implementar las técnicas que se describirán a lo largo de esta sección, con lo que se justificará la selección del mismo.

Se seleccionaron cables de *Ethernet* de diferentes longitudes para la toma de resultados que se mostrarán en el capítulo 5. Los cables de *Ethernet* de categoría 6 poseen características y especificaciones para evitar la diafonía y minimizar el ruido, siendo el estándar de cable que se utiliza para comunicaciones 10BASE-T, 100BASE-TX y 1000BASE-TX. Es capaz de alcanzar frecuencias de hasta 250MHz, lo cual se acomoda perfectamente a los requerimientos de la señal de reloj que enviaremos entre placas. De los cuatro pares que constituyen dicho cable se ha utilizado exclusivamente un par durante la fase experimental. Como todos los cables definidos por los estándares TIA/EIA-568-B, se indica la máxima longitud horizontal de un cable de categoría 6, para poder aprovechar sus características y asegurar la llegada de la señal de manera correcta en el receptor, siendo esta de 90 metros [16]. Los valores propuestos por IEEE que se deben cumplir para las frecuencias expuestas en la siguiente tabla son:

| Frecuencia(MHz) | Atenuación(dB) | Retraso de fase (ns) |
|-----------------|----------------|----------------------|
| 100 | 21,7 | 549,4 |
| 200 | 31,7 | 548,4 |
| 250 | 36,0 | 548,2 |

TABLA 4.1: Características cable *Ethernet* categoría 6.

Los cables de categoría 6 que se han incluido para la comunicación de ambas placas de desarrollo son de tipo UTP (*Unshielded Twisted Pair*) como se muestra en la figura 4.2. Se encuentra normalizado de acuerdo a la norma estadounidense TIA/EIA-568-B y a la internacional ISO/IEC 11801. Su impedancia característica es de 100Ω .

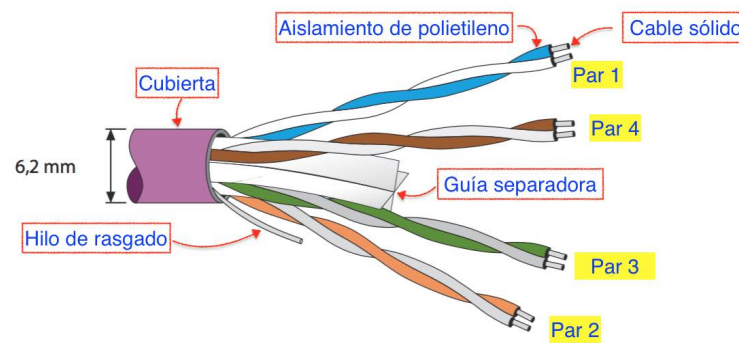


FIGURA 4.2: Cable UTP categoría 6.

4.2.1. Diafonía. Trenzado de cables

El trenzado de cables es una técnica usada en telecomunicaciones que consiste en entrelazar dos cables de manera helicoidal para reducir e incluso anular interferencias de fuentes externas y diafonía producida por la circuitería adyacente. Esto se debe a que dos cables paralelos se comportan como una antena simple. En los cables de *Ethernet* los dos hilos del par se trenzan entre sí de forma que el campo magnético generado por cada hilo se cancela con el de su par, lo que lo protege de interferencias exteriores y hace que la emisión de señal a otros pares cercanos sea menor, permitiendo así una mejor transmisión de datos. Esta técnica permite reducir la interferencia eléctrica procedente de diversas fuentes reduciendo así el ruido sobre la señal transmitida. A la hora de transmitir una señal, si ésta es a bajas frecuencias, la técnica de trenzado permite a los cables eliminar la mayor parte de los efectos del ruido electromagnético sobre la señal, mientras que en altas frecuencias esos efectos son evitados por el blindaje del cable en caso de que éste lo posea.

Como se verá avanzado este capítulo, será necesario la emisión de señales en modo diferencial para reducir el ruido, el entrelazado de cables que llevan señales en modo diferencial presenta también una serie de ventajas. Si tenemos que la amplitud de onda que viaja por uno de los cables es $A(t)$ y en el otro es $-A(t)$, y $n(t)$ es ruido añadido por igual en ambos cables, tendremos: $A(t) + n(t)$ en un cable y en el otro $-A(t) + n(t)$; al hacer la diferencia en el receptor se habrá eliminado el ruido. Además, el campo magnético que producirá la corriente que circula por uno de ellos, será de sentido opuesto al que produce la corriente de que circula en sentido opuesto en el otro par, con lo que se eliminarán los campos fuera del cable, evitando así que se induzca alguna corriente en circuitos cercanos.

Sin embargo, este método no puede ser implementado exactamente así, el núcleo de las FPGA de Xilinx no puede trabajar con intensidades de corriente negativas, con lo cual no son emitidas corrientes diferenciales de esta manera en la placa Basys 3.

4.2.1.1. Atenuación

La atenuación de una señal eléctrica consiste en la pérdida de potencia sufrida por la misma al transitar por cualquier medio de transmisión a una determinada frecuencia. Si introducimos una señal eléctrica con una potencia P_2 en un circuito pasivo, como puede ser un cable, esta sufrirá una atenuación y al final de dicho circuito poseerá una potencia P_1 . La atenuación α será igual a la diferencia entre ambas potencias. La atenuación en función de la potencia viene dada por la siguiente expresión:

$$\alpha = 20 \log \frac{P_1}{P_2} \quad (4.1)$$

La atenuación es un factor limitante al seleccionar la longitud del cable utilizado dependiendo de la frecuencia que se quiera transmitir[17]. Entre los factores de los que depende la atenuación en estos cables destacan la temperatura y la frecuencia de la señal. La atenuación se expresa en dB/Km y su relación con la frecuencia viene dada por la siguiente expresión:

$$\alpha = 0,01 + 2,3\sqrt{f} + 0,003f \quad (4.2)$$

Debido a los valores de atenuación que presentan los cables de *Ethernet* de categoría 6 para los valores de frecuencia que se muestran en la tabla 4.1, se puede justificar también la inclusión de este cable con longitudes de 5 m, 10 m, 20 m para las correspondientes pruebas, puesto que una pérdida excesiva en la potencia de la señal transmitida por el cable, en este caso la señal de reloj, puede provocar una interpretación errónea en el receptor, lo que se traduce como la pérdida de pulsos en el receptor al no interpretar correctamente los voltajes de la señal.

4.2.1.2. AWG

AWG (*American Wire Gauge*) es una referencia de clasificación de diámetros. Cuanto más alto es su valor, más delgado es el alambre que constituye el núcleo del cable al que se refiere. Un mayor grosor, es decir un AWG más bajo, posee menos resistencia interna y, por lo tanto, soporta valores de corriente mayores a distancias más grandes. El AWG del cableado seleccionado para llevar a cabo las pruebas posteriores es de 24 y sus características se pueden observar en la tabla 4.2.

| AWG | Diámetro(mm) | Resistencia eléctrica (Ω/Km) |
|-----|--------------|---------------------------------------|
| 24 | 21,7 | 549,4 |

TABLA 4.2: Valores de AWG 24.

4.3. Mejoras sobre elementos activos. Señales diferenciales

La señalización diferencial consiste en el envío de una señal por dos conductores de tal manera que los voltajes e intensidades en ellos son simétricos. El valor útil de señal se obtiene restando el voltaje de ambos conductores: $V_+ - V_-$. Mediante el uso de señales construidas de manera

diferencial se consigue una mayor inmunización de la señal frente al ruido electromagnético. Esto se debe a que a la hora de calcular el valor del voltaje efectivo, en el caso de presentarse ruido, éste se anula, ya que la interferencia se suma por igual en ambos conductores. Como ya se mencionó en el capítulo 2, el núcleo de las FPGA de Xilinx es incapaz de trabajar con voltajes de polarización negativa, por lo que se sirve de una serie de elementos que se describen a continuación para el manejo y construcción de señales diferenciales, contruidos con valores de tensión positivos los cuales vienen determinados por los valores V_{Ref} y V_{CC0} que aparecen en la figura 4.3.

4.3.1. Voltajes asociados a los IOB

Las señales emitidas desde los pines de salida de la FPGA tienen un voltaje asociado, los cuales son especificados en los archivos de restricciones incorporados al proyecto, con extensión .xdc en el caso de utilizar el *software* Vivado y extensión .ucf en el caso de utilizar EDK, cambiando la sintaxis asociada de uno respecto al otro. Los pines de salida de las FPGA están distribuidos en bancos de salida, como se indicó en el capítulo 2. Se puede variar el voltaje asociado de estos bancos desde los archivos de restricciones, pero al cambiar el voltaje de un banco a uno de los admisibles se debe cambiar también el voltaje de todos los pines conectados a ese banco que vayan a ser utilizados en el diseño, y por ende en los archivos de restricciones habrá que prestar especial atención a este aspecto, seleccionando voltajes compatibles entre sí para todos los pines afectados. A la hora de seleccionar voltajes compatibles se tendrá en cuenta el voltaje V_{CC0} y los voltajes de referencia V_{REF} . Debido a la intención de utilizar señales diferenciales y que los IOB de la FPGA Artix 7 con el empaquetamiento 1CPG236C son todos de tipo HR (*High Range*), se seleccionó el estándar DIFF_HSTL_I_18 para los pines asociados a la salida del reloj diferencial en el circuito emisor y los de entrada en el circuito receptor. Dicho estándar es compatible con el HSTL_I_18, que deberá estar asociado al resto de pines utilizados que compartan el mismo banco, cuyo V_{CC0} es de 1.8 v y el voltaje de referencia (V_{REF}) de 0.9 v como se aprecia en la figura 4.3. Además, estos estándares deben ser compatibles con los elementos de diseño que se utilizarán para la generación y recepción del reloj diferencial que serán descritos a continuación.

| I/O Standard | I/O Bank Availability | V_{CC0} (V) | | | V_{REF} (V) |
|----------------|-----------------------|---------------|-------|--------------------------------|---------------|
| | | Output | Input | Input with DIFF_TERM = TRUE | Input |
| DIFF_HSTL_I_18 | Both | 1.8 | Any | N/A | N/A |
| HSTL_I_18 | Both | 1.8 | Any | N/A | 0.9 |

FIGURA 4.3: Requisitos VCCO y VREF para los estándares seleccionados. Figura cortesía de [11].

4.3.2. Buffers diferenciales. Amplificadores operacionales

Una señal binaria se puede enviar en modo diferencial sobre un par de cables al establecer uno de los conductores con la señal y un segundo hilo con el complemento de la señal. En el envío de un 1 lógico, se transmite un voltaje V_1 por una de las líneas del par, y un voltaje V_0

por la otra línea del par. Para enviar un 0, se invierten los voltajes. Cuanto mayor es la diferencia entre estos dos voltajes, mayor inmunización al ruido se consigue. En la construcción del reloj diferencial en el desarrollo de este TFG se han añadido los componentes de diseño que se pasan a describir a continuación, y que han sido incluidos en el diseño VHDL tanto de los circuitos emisor como el receptor. El comportamiento de estos componentes se basa en el funcionamiento de amplificadores operacionales y construirán los voltajes de señal en el receptor en función a las propiedades de éstos. Estos dispositivos poseen dos entradas: una entrada llamada no inversora V_1 , y otra inversora sometida a una tensión V_2 como se aprecia en la figura 4.4. En forma ideal, el dispositivo amplifica solamente la diferencia de tensión en las entradas, conocida como tensión de entrada diferencial. La tensión de salida del dispositivo V_{out} viene dada por la ecuación:

$$V_{out} = A_{OL}(V_1 - V_2) \quad (4.3)$$

A_{OL} representa la ganancia del dispositivo en en lazo abierto y toma valores muy elevados, determinados en el proceso de fabricación. En algunos amplificadores diferenciales, existen dos salidas con desfase de 180 grados para algunas aplicaciones especiales.

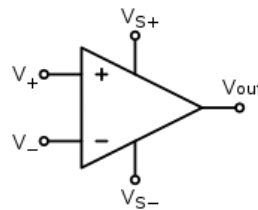


FIGURA 4.4: Símbolo del amplificador operacional.

4.3.2.1. OBUFDS

Este elemento de diseño consiste en un *buffer* de salida único que admite señales diferenciales de bajo voltaje. OBUFDS aísla el circuito interno y proporciona una corriente para las señales que salen de la FPGA. Su salida se representa como dos puertos distintos (O y OB), uno se considera el maestro y el otro el esclavo. El maestro y el esclavo poseen fases opuestas de la misma señal lógica [11]. Este elemento ha sido instanciado en el diseño VHDL del circuito emisor de señales, para, de esta manera, emitir dos señales con fase opuesta que saldrán de la FPGA y constituirán la señal de reloj diferencial. A continuación, se adjunta la plantilla proporcionada por Xilinx [18] para crear una instancia de este componente en el diseño del circuito emisor:

```
-- OBUFDS: Differential Output Buffer
-- 7 Series
-- Xilinx HDL Libraries Guide, version 13.4
OBUFDS_inst : OBUFDS
generic map (
  IOSTANDARD => "DEFAULT", -- Specify the output I/O standard
  SLEW => "SLOW") -- Specify the output slew rate
port map (
```

```
O => O, -- Diff_p output (connect directly to top-level port)
OB => OB, -- Diff_n output (connect directly to top-level port)
I => I -- Buffer input
);
-- End of OBUFDS_inst instantiation
```

El atributo SLEW nos permite minimizar el efecto con el mismo nombre producido por los amplificadores al generar las señales y que se describe en las siguientes subsecciones.

4.3.2.2. IBUFDS

Este elemento de diseño consiste en un *buffer* de entrada que admite señalización diferencial de bajo voltaje, y que generará, a partir del voltaje de dos señales de entrada y mediante la inclusión de un amplificador operacional, la señal con el voltaje asociado resultado de esta amplificación. En un IBUFDS, una señal de interfaz de nivel de diseño se representa como dos puertos distintos (I e IB), uno considerado el maestro y el otro el esclavo. Opcionalmente, está disponible el uso de una resistencia de terminación diferencial programable para ayudar a mejorar la integridad de la señal y así reducir la necesidad de añadir los componentes externos que llevan a cabo esta función. Sin embargo, con el estándar de señalización utilizado en las placas Basys 3 no es posible su uso. Este componente ha sido instanciado en el diseño VHDL del circuito receptor de señales, utilizando una la plantilla proporcionada por Xilinx, para de esta manera recibir dos señales con fase opuesta y construir a partir de éstas la señal que gobierna dicho diseño. A continuación, se adjunta la plantilla proporcionada:

```
-- IBUFDS: Differential Input Buffer
-- 7 Series
-- Xilinx HDL Libraries Guide, version 13.4
IBUFDS_inst : IBUFDS
generic map (
DIFF_TERM => FALSE, -- Differential Termination
IBUF_LOW_PWR => TRUE, -- Low power (TRUE) vs. performance (FALSE)
IOSTANDARD => "DEFAULT")
port map (
O => O, -- Buffer output
I => I, -- Diff_p buffer input (connect directly to top-level port)
IB => IB -- Diff_n buffer input (connect directly to top-level port)
);
-- End of IBUFDS_inst instantiation
```

El atributo DIFF_TERM permite la utilización de una resistencia de terminación cuya función se describirá más avanzado este capítulo.

4.3.3. Terminación del cable

A menudo se piensa en los cables como conductores ideales que poseen cero resistencia, capacitancia e inductancia. Sin embargo, un cable real no es un conductor ideal, sino un elemento de circuito pasivo, y a medida que la frecuencia de las señales que viajan por ellos aumenta esto se hace más evidente. A la hora de realizar el diseño se deben tener en cuenta estas propiedades

eléctricas del cable que se usa para llevar a cabo la comunicación entre ambas FPGA. Todas estas propiedades dependen en gran medida de la geometría de dicho cable:

- La resistencia es la oposición que ofrece el cable como elemento conductor al paso de una corriente. La resistencia de un conductor es proporcional a su longitud dividida por su área eléctrica y también depende de la frecuencia, este valor de resistencia es muy reducido.
- la capacitancia es la capacidad que tiene el cable, al ser un conductor, para almacenar carga eléctrica. La capacidad entre dos conductores depende de su entorno global, ya que los conductores intermedios o cercanos pueden crear o desviar líneas de campo. Además, el elemento capacitivo en una línea también se ve influenciado por la existencia de una señal de retorno estableciéndose así una relación entre la señal y el retorno que se produce. La capacitancia depende también de la permitividad del material aislante, que actúa dieléctrico. En general, es ventajoso usar un material aislante con la permitividad más baja posible, ya que proporciona no solo una capacitancia menor sino también una velocidad de propagación más alta.
- La inductancia es la propiedad de un circuito eléctrico para resistir el cambio de corriente. Una corriente que fluye a través de un cable genera un campo magnético alrededor. Dicho campo magnético genera una fuerza contraelectromotriz que se opone al paso de la corriente.
- La impedancia característica es una propiedad que presenta una línea de transmisión considerada infinita. Se puede ver como la relación que existe entre el voltaje aplicado y la corriente absorbida por la línea en el caso de que tenga una longitud infinita, o que no presente corriente de reflexión. Intervienen en su valor los valores de resistencia, inductancia, capacidad que se acaban de describir y también la resistencia del aislamiento. Cabe destacar que, a la hora de comunicar diferentes circuitos, es aconsejable que tanto el circuito emisor como el receptor y el cable que los comunique presenten las mismas impedancias, para así minimizar las pérdidas en la señal que esto puede suponer.

Aunque los superconductores pueden reducir la resistencia a cero, a un coste considerable, la inductancia y la capacitancia no pueden ser eliminadas. Teniendo en cuenta esto, los cables reales tienen por tanto resistencia e inductancia en serie y capacitancia paralelas. Es importante tener en cuenta que los elementos del circuito de un cable se distribuyen a lo largo de su longitud y no se agrupan en una sola posición como se puede apreciar en la figura 4.5.

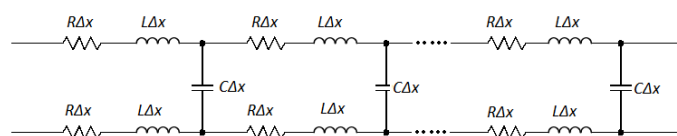


FIGURA 4.5: Resistencia, inductancia y capacitancia de un cable no ideal.

4.3.3.1. Señal reflejada

En general, la solución para las corrientes y las tensiones de un circuito formado por resistencias, condensadores e inductancias, y sin ningún componente de comportamiento no lineal, supone resolver ecuaciones diferenciales. Pero, cuando todos los generadores de tensión y de corriente tienen la misma frecuencia constante y sus amplitudes son constantes, las soluciones en estado estacionario, es decir cuando todos los fenómenos transitorios han desaparecido, son sinusoidales y todas las tensiones y corrientes tienen la misma frecuencia que los generadores y amplitud constante. Un cable infinito ideal al que se le aplica una diferencia de potencial, que cambia de manera instantánea, y que presenta una capacitancia provoca la aparición de una corriente sobre él. Dado que el cable es infinito y por tanto su capacitancia también, esta corriente estará siempre presente. A la par, debido a la inductancia del cable, se genera una resistencia al paso de dicha corriente. Surge así el concepto de impedancia característica del cable Z , como una generalización de la ley de Ohm que se aprecia en la siguiente expresión, como la relación entre la diferencia de potencial aplicado y la corriente que surge:

$$Z = \frac{V}{I} \quad (4.4)$$

Como se ilustra en la figura 4.5, una línea de transmisión se puede representar gráficamente como una red de cuatro puertos. La figura muestra una línea que ha sido discretizada en secciones cortas de una cierta longitud. Cada sección está modelada por un circuito que consta de una resistencia (R), una inductancia (L) y una conductancia (C) que se distribuyen a lo largo del cable. Cuando se tiene un cable finito al que se le aplica una diferencia de potencial que cambia instantáneamente, como es el caso de la señal de reloj, el circuito receptor va a reflejar parte de la energía de la señal que le llega. La ruta superior en la figura 4.5 representa la ruta de la señal al viajar desde el emisor al receptor, mientras que la ruta inferior representa la ruta que sigue la señal reflejada, y que se debe eliminar utilizando una resistencia de terminación, simulando así la adición de un cable infinito [12]. Como se verá en el capítulo 5, esta señal reflejada puede producir un mal comportamiento, tanto por la aparición de una corriente de retorno como por la posibilidad de aparición de ondas estacionarias en la señal de reloj transmitida entre los diseños que se han realizados y, por consiguiente, debe ser eliminada.

4.3.3.2. Resistencia de terminación

Mediante la utilización de una resistencia de terminación R_T se consigue simular el comportamiento de un cable infinito, al absorber la señal de retorno que el circuito receptor refleja. Esta resistencia debe tener un valor igual al de la impedancia característica del cable. Si el valor de la resistencia terminal difiere del de la impedancia característica, la corriente de retorno será proporcional al grado de desigualdad entre ellas. Para una correcta transmisión, además, la impedancia de la fuente que excita la línea, en este caso el circuito emisor, debe también tener una impedancia igual a la impedancia característica del cable. Dichas resistencias terminales se pueden colocar tanto cerca del emisor como del receptor de la señal y se basan en la aplicación de las leyes de

Kirchhoff para eliminar la señal de retorno en el cable. En la figura 4.6 se muestra la manera clásica de colocar una resistencia de terminación utilizando señalización diferencial.

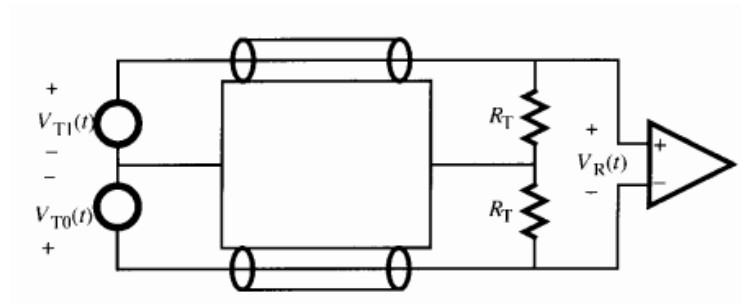


FIGURA 4.6: Diseño de terminación para señalización diferencial. Figura cortesía de [12].

Con señalización diferencial bipolar, $V_1 = -V_0$, la corriente de retorno siempre es cero tras añadir una resistencia de terminación, como se puede deducir de la ecuación 4.5. Se debe recordar que en el caso de las placas Basys 3 utilizadas, la señalización diferencial se construye con voltajes mayores de cero.

$$I_R = \frac{V_1 + V_0}{R_T} \quad (4.5)$$

En la figura 4.7 se puede apreciar el diseño de terminación propuesto por Xilinx para los estándares de señalización utilizados.

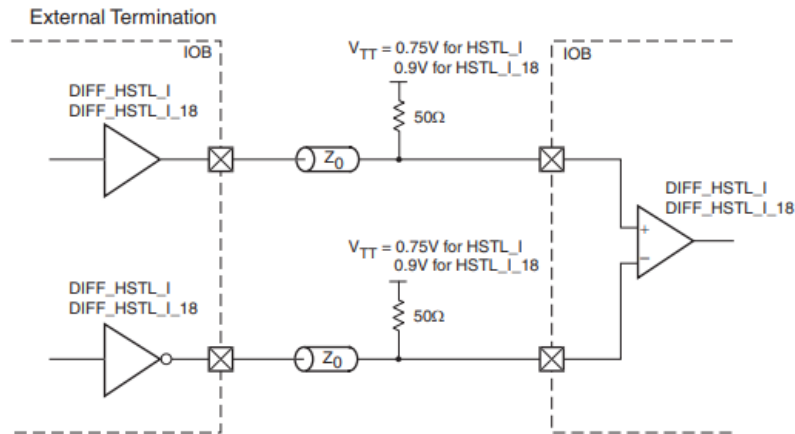


FIGURA 4.7: Diseño de terminación propuesto por Xilinx. Figura cortesía de [11].

4.4. Efectos jitter, skew y slew rate

El *jitter* es un efecto que consiste en la variación en el tiempo de una señal digital durante el envío de ésta, y está considerado como ruido no deseado en la señal. El *jitter* se produce porque el transcurso real esperado de la señal varía respecto al transcurso ideal esperado de la señal en el área alrededor de los flancos de subida y de bajada, como se puede observar en la figura 4.8. Esto

podría provocar comportamientos no deseados en el circuito receptor, lo que se traduce como un retraso de la señal.

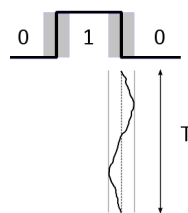


FIGURA 4.8: Efecto *jitter*.

El sesgo en la señal de reloj (*skew*) se produce cuando una señal de reloj no llega a todos los elementos a la vez. Esto puede suceder tanto porque el material que conduce la señal no es el ideal, provocando que circule más rápido o despacio de lo deseado, como porque la distancia desde el reloj hasta cada componente no es la misma, y la señal tarda más en recorrer caminos más largos. Esto puede causar problemas en el circuito, ya que algunos componentes, como los biestables, necesitan tener los datos en la entrada justo en el momento en el que son activados por el reloj, y no antes ni después. Justamente es por esto por lo que las FPGA distribuyen la señal de reloj como se vio en el capítulo 2.

Además, para eliminar estos efectos en relojes procedentes del exterior, las FPGA, usadas para implementar los diseños, cuentan con un PLL y MMCM por cada CMT, como se vio en el capítulo 2, los cuales están precisamente destinados a eliminar los efectos *jitter* y *clock skew* mencionados en esta subsección, entre otras funciones.

Se utiliza la restricción `CLOCK_DEDICATED_ROUTE=FALSE` para indicar a Vivado y EDK que el CCIO puede redireccionarse a un PLL de una región adyacente durante la fase de síntesis del circuito RTL, en caso de ser necesario.

El SR (*Slew Rate*) es un efecto que consiste en la incapacidad de los amplificadores para seguir variaciones rápidas de la señal de entrada. Se define como la máxima tasa de cambio en el voltaje de salida cuando el voltaje de entrada cambia. Es importante asignar al atributo `SLEW` del IBUFDS implementado el valor `SLOW` para un correcto comportamiento ante este efecto, de esta forma se indica al diseño del circuito bajo prueba un cambio lento en los flancos de subida y bajada y así se consigue una mejor interpretación de la señal en el receptor.

4.5. Limitaciones encontradas

Durante la fase de diseño de los circuitos emisores y receptores que implementaban los componentes para la emisión y recepción de las señales diferenciales se intentó generar dichas señales diferenciales mediante los estándares: LVDS_25 y LVCMOS25. Posteriormente se comprobó que los PMOD incluidos en la placa Basys 3 de Diligent no eran compatibles con los voltajes V_{CCO} y V_{REF} asociados a este estándar de señalización, por lo que se tuvieron que generar las señales con los estándares de señalización DIFF_HSTL_I_18 y HSTL_I_18, compatibles entre sí, para poder

usar los PMOD.

4.5.1. Conectores SMA. Placa XUPV5-LX110T

El cable coaxial es un cable utilizado para transportar señales eléctricas de alta frecuencia formado por conductores concéntricos, uno central, llamado núcleo, encargado de llevar la información, y uno exterior, de aspecto tubular, llamado malla, que sirve como referencia de tierra y retorno de las corrientes. Mientras que el núcleo de un cable coaxial transporta señales eléctricas que constituyen la información, la malla de hilo trenzada que lo rodea actúa como masa, protegiendo al núcleo del ruido eléctrico y de la distorsión que proviene de los cables adyacentes y del espacio exterior.

El cable coaxial, al no verse afectado por interferencias externas, es capaz de lograr altas velocidades de transmisión de datos en largas distancias. Por esta razón, se ha pensado en él como elemento pasivo para inmunizar contra la radiación, como alternativa al cable trenzado de *Ethernet*.

Ante la ausencia de conectores para el uso de este tipo de cables en la placa Basys 3, y la imposibilidad de generar voltajes superiores con otros estándares, se pensó en la utilización de una placa de desarrollo XUPV5-LX110T5 con una FPGA Virtex 5 empotrada en ella. Como se vio en el capítulo 2 la placa XUPV5-LX110T cuenta con 8 conectores SMA.

Cuatro de estos conectores permiten emitir y recibir, desde y hacia la FPGA, señales de reloj diferenciales a través de ellos[7] y poseen una impedancia de 50Ω . La Virtex 5 puede configurarse con una impedancia de terminación de 100Ω al activar el atributo DIFF_TERM en la instancia del componente IBUFDS. Se puede encaminar una señal de reloj diferencial a través de un multiplexor de reloj LVDS a un segundo par de conectores SMA. Esto permite el envío de un reloj de precisión a un dispositivo externo, que en este caso será el circuito bajo prueba, también implementado sobre otra placa XUPV5-LX110T.

Al igual que se hizo con la placa Basys 3, se generará el reloj diferencial mediante la instancia del componente OBUFGDS y conectando sus salidas a los pines asociados a los conectores SMA. De igual manera el circuito receptor implementado en otra placa XUPV5-LX110T recibirá el reloj diferencial por sus conectores SMA y construirá el reloj que gobierna el diseño a partir de éstos, mediante la inclusión en el diseño de una instancia del componente IBUFDS.

En este caso, los IOB a los que pertenecen tanto los pines de salida conectados a los conectores SMA en el circuito emisor, como los del circuito receptor, utilizarán los estándares de entrada/salida LVDS_25 y LVCMOS25. El voltaje V_{CCO} asociado a estos estándares es de 2.5v. Hay que recordar que cuanto mayor es la diferencia de voltajes utilizados para construir la señal diferencial, mayor inmunización al ruido se consigue.

Ambas placas XUPV5-LX110T se programan mediante Impact, como se indicó en el capítulo 2, usando un conector JTAG.

Capítulo 5

Resultados

En este capítulo se exponen los resultados obtenidos en las mediciones asociadas a cada una de las mejoras realizadas sobre los circuitos iniciales y las líneas de comunicación que los conectan. Se medirán los pulsos perdidos entre circuitos emisor de ciclos de reloj y receptor de estos.

5.1. Resultados del diseño original

Se partió de un diseño que interconectaba el circuito emisor y receptor mediante un par trenzado sencillo, compuesto con dos cables de cobre, que unía el diseño emisor con el receptor. Tras configurar las placas Basys 3 con los diseños, se procedió a enviar una señal binaria común de reloj de 100MHz, conectando las masas de ambas placas entre sí. El estándar de señalización utilizado en este caso ha sido LVCMOS33, cuyo V_{CC0} es de 3.3v. Los resultados fueron los siguientes, tras repetir los envíos un total de 10 veces para cada caso:

| Longitud del cable (cm) | Ciclos de reloj emitidos | Ciclos de reloj perdidos |
|-------------------------|--------------------------|---|
| 16 | 2^{32} | 0 |
| 32 | 2^{32} | De $2^{32} - 6$ a $2^{32} - 4$ aleatorios |
| 48 | 2^{32} | De $2^{32} - 2$ a $2^{32} - 1$ aleatorios |

TABLA 5.1: Resultados en la transmisión de una señal de reloj digital a 100MHz.

Tras la toma de resultados, cuyos valores se observan en la tabla 5.1, se llegó a las siguientes conclusiones:

- Que enviando hasta un total de 2^{32} ciclos con cables de longitud menor a 32cm rara vez había pérdida en la recepción de la señal.
- Al aumentar la longitud de par trenzado, que separaba ambos diseños, el número de pulsos perdidos era aleatorio.
- Al aumentar la longitud del cable también aumentaba el número de ciclos perdidos, pero aun siendo aleatorio este número, cada vez era mayor esta pérdida.

5.2. Resultados con señales diferenciales

En esta subsección se plasman los resultados del envío del reloj en modo diferencial. Como se dijo en el capítulo 4 se añadieron los componentes IBUFDS y OBUFGDS a ambos circuitos implementados sobre placas Basys 3.

Con una frecuencia de reloj de 100MHz se consiguió transmitir la señal de manera fiable a través de uno de los pares trenzados que constituyen un cable de *Ethernet* de categoría 6, de longitud 1m, cuyas características se explicaron en el capítulo 4. Las pruebas se basaron en el envío de una señal de reloj de hasta un total 2^{38} ciclos consecutivos. Esta cantidad de ciclos está relacionada con el tiempo de cada prueba, así 2^{38} ciclos a una frecuencia de 100MHz supone un tiempo de 46 minutos aproximadamente. Al modificar la longitud del cable a 5m, 10m y 20m, manteniendo el reloj generado a una frecuencia de 100MHz, se obtuvieron los resultados expuestos en la tabla 5.2. La cantidad de ciclos de reloj perdidos fue constante para cada longitud de cable, independientemente del número de ciclos enviados desde la placa emisora. Se enviaron hasta un total de 2^{38} ciclos de reloj.

| Longitud del cable (m) | Ciclos de reloj emitidos | Ciclos de reloj perdidos |
|------------------------|--------------------------|--------------------------|
| 1 | 2^{38} | 0 |
| 5 | 2^{38} | 5 |
| 10 | 2^{38} | 11 |
| 20 | 2^{38} | 21 |

TABLA 5.2: Resultados en la transmisión de una señal de reloj en modo diferencial a 100MHz.

Debido a que en las líneas de transmisión no se utilizó una resistencia de terminación para simular el comportamiento de un cable infinito, en los resultados obtenidos se puede ver que el número de pulsos perdidos es fijo, debido al rebote de parte de la señal a través de la línea. También se puede ver que el estándar de señalización utilizado es muy sensible al ruido. Aun así, se ha conseguido transmitir de manera fiable la señal a una distancia de 1m, aumentando la distancia en comparación con el diseño original en 68cm. Las pruebas se repitieron un total de 10 veces con idénticos resultados.

5.3. Resultados añadiendo una resistencia de terminación

Tras observar que los resultados de las pérdidas fueron constantes, se procedió a visualizar con un osciloscopio los voltajes asociados que constituirían la señal de reloj emitidas desde el circuito emisor. De esta manera se pudo apreciar el reflejo de la señal, por lo que impulsado por el afán de aumentar la frecuencia y longitud a la que se transmitía el reloj de una manera fiable, y ante la imposibilidad de usar el atributo DIFF_TERM en los diseños de la placa Basys 3, como se explica en el capítulo 4, se añadió una resistencia de terminación externa de 100Ω .

En las mediciones con reloj emitido en modo diferencial a una frecuencia de 100MHz, para las longitudes de cable de *Ethernet* de 5m, 10m y 20m se consiguió transmitir el reloj de manera

fiable sin pérdidas en el receptor, emitiéndose y recibándose hasta un total de 2^{38} ciclos de reloj. como en casas anteriores las pruebas se repitieron un total de 10 veces.

Los resultados obtenidos en esta fase aparecen en la tabla 5.3.

| Frecuencia (MHz) | Longitud del cable (m) | Ciclos de reloj emitidos | Ciclos de reloj perdidos |
|------------------|------------------------|--------------------------|--------------------------|
| 100 | 5 | 2^{38} | 0 |
| 100 | 10 | 2^{38} | 0 |
| 100 | 20 | 2^{38} | 0 |
| 110 | 5 | 2^{38} | 1 |
| 110 | 10 | 2^{38} | 1 |
| 110 | 20 | 2^{38} | Entre 1 a 7 aleatorios |
| 125 | 5 | 2^{38} | 1 |
| 125 | 10 | 2^{38} | 1 |
| 125 | 20 | 2^{38} | Entre 1 a 8 aleatorios |
| 150 | 5 | 2^{38} | 1 |
| 150 | 10 | 2^{38} | Entre 1 - 8 aleatorios |
| 150 | 20 | 2^{38} | $2^{38} - 9$ |

TABLA 5.3: Resultados en la transmisión de una señal de reloj en modo diferencial para diferentes frecuencias sobre placas Basys 3.

Se puede observar en estos resultados que la máxima frecuencia a la que se consiguió emitir el reloj de manera diferencial, utilizando una resistencia de terminación fue de 100MHz. Como se puede ver en la tabla 5.3, al ir aumentando la frecuencia de reloj y la longitud del cable se reduce el número de ciclos que se recibe, siendo estos un total de 9 a una longitud de 20m y una frecuencia de 200MHz, independientemente del número que sea transmitidos desde el emisor. En los resultados obtenidos podemos observar que el aumento de la frecuencia junto con la longitud del cable provoca errores en la recepción. Debido a la cantidad de ciclos obtenidos es fija cuando se produce error, se piensa que esto está producido por el reflejo de la señal. Como se vio en el capítulo 4 el reflejo de la señal está asociada a la aparición de ondas estacionarias a lo largo de las líneas de comunicación. Además, se debe recordar que, para un correcto funcionamiento, las impedancias de los circuitos emisor y receptor y la impedancia característica del cable deben de ser la misma, y debido a los elementos utilizados en este estudio esto no es así. Todo esto provocaría una alteración en la señal transmitida y una mala interpretación en la recepción, como es el caso. A pesar de todo ello, se ha conseguido aumentar la distancia, a la que se transmite la señal de manera fiable, sin reducir la frecuencia, en 19,68m para una frecuencia de 100MHz.

5.4. Resultados usando placas XUPV5-LX110T y conectores SMA

Por último, se decidió realizar las pruebas anteriores sobre dos placas XUPV5-LX110T, debido a que la señal diferencial generada desde la placa emisora utiliza un circuito externo a la FPGA, el cual puede trabajar con el estándar de señalización LVDS_25. Tampoco fue necesario incorporar una resistencia de terminación, pues activando el atributo DIFF_TERM, se utiliza una de 100Ω

integrada en la placa, como se vio en el capítulo 4. En la tabla 5.4 se pueden observar los resultados obtenidos nuevamente con diferentes frecuencias de reloj y longitudes de cable de *Ethernet* de categoría 6. Las pruebas se repitieron una vez más un total de 10 veces, con los resultados que se pueden ver en la tabla 5.4.

| Frecuencia (MHz) | Longitud del cable (m) | Ciclos de reloj emitidos | Ciclos de reloj perdidos |
|------------------|------------------------|--------------------------|--------------------------|
| 100 | 5 | 2^{38} | 0 |
| 100 | 10 | 2^{38} | 0 |
| 100 | 20 | 2^{38} | 0 |
| 150 | 5 | 2^{38} | 0 |
| 150 | 10 | 2^{38} | 0 |
| 150 | 20 | 2^{38} | 0 |
| 170 | 5 | 2^{39} | 0 |
| 170 | 10 | 2^{39} | 0 |
| 170 | 20 | 2^{39} | 0 |
| 175 | 5 | 2^{39} | 0 |
| 175 | 10 | 2^{39} | 0 |
| 175 | 20 | 2^{39} | 0 |
| 200 | 5 | 2^{39} | Entre 3 - 9 aleatorios |
| 200 | 10 | 2^{39} | 0 |
| 200 | 20 | 2^{39} | $2^{39} - 7$ |

TABLA 5.4: Resultados en la transmisión de una señal de reloj en modo diferencial a diferentes frecuencias sobre placas XUPV5-LX110T.

Como se observa en la tabla 5.4 se ha conseguido emitir de manera fiable la señal de reloj en modo diferencial a una frecuencia de reloj de 170MHz a una distancia de 20m; y a una frecuencia de 200MHz con un cable de 10m de longitud. Estos resultados son sustancialmente mejores si los comparamos con los obtenidos con las placas Basys 3. En gran parte esto se debe a la posibilidad de instanciar en el diseño una resistencia de terminación, presente en la placa. De esta manera se evita la utilización de una resistencia externa y otros elementos pasivos para conectarla, los cuales perjudican la transmisión debido a sus imperfecciones. También, gracias a una mayor diferencia entre los voltajes utilizados para la construcción de la señal, aumenta la inmunización al ruido, consiguiendo así mejores resultados. Al igual que ocurrió al analizar los resultados de la placa Basys 3, se puede observar, que nuevamente el aumento de la frecuencia junto con la longitud del cable provoca errores en la recepción.

Por tanto, al finalizar la toma de resultados, se puede afirmar que se ha logrado aumentar la frecuencia a la que se emite la señal de reloj de manera fiable a una distancia de 20m, a un total de 170MHz. También se ha conseguido aumentar la frecuencia en un 100 % si comparamos los resultados obtenidos con los que se obtuvieron con la placa Basys 3, es decir a 200MHz, para una distancia de 10m y de manera fiable.

Capítulo 6

Conclusiones

Con el fin de bombardear con neutrones de alta velocidad diversas FPGA, con un acelerador de partículas, para someter a estudio su comportamiento, y evaluar la calidad de las técnicas de protección empleadas en los diseños que implementan frente a los SEU, surgió la necesidad de generar de una manera externa y totalmente controlada la señal de reloj que gobierna el sincronismo en dichos diseños. De esta manera es posible forzar un estado esperado en los circuitos bajo prueba y compararlos con los obtenidos durante el experimento. Es así como surge este TFG el cual está enmarcado en el contexto descrito.

A la hora de transmitir dicha señal de manera externa surgen problemas en la recepción y transmisión de esta de manera fiable. Debido al ruido electromagnético e impulsivo presente en estos entornos, generado en gran parte por el propio acelerador de partículas, así como por la alimentación de los dispositivos involucrados, es necesario inmunizar dicha señal ante éstos. En este TFG este problema se ha abordado mediante la transmisión de señales de manera diferencial y utilizando técnicas y elementos capaces de reducir el acoplamiento de ruido en ella.

Gracias a la generación de la señal de reloj de manera diferencial, se ha logrado transmitir de forma fiable la señal de reloj entre dos FPGA Virtex 5 a una frecuencia de 170MHz, utilizando para ello uno de los pares trenzados que constituyen un cable de *Ethernet* UTP de categoría 6 hasta 20m de longitud. Esta señal se ha transmitido a través de dos conectores SMA integrados en una placa de desarrollo XUPV5-LX110T y se han generado con un circuito externo a la FPGA presente en la placa. Debido a la posibilidad de instanciar una resistencia de terminación presente en la placa, que evita el uso de soldaduras y resistencias externas, y el rango de voltajes con los que se construye la señal diferencial, se han obtenido mejores resultados que utilizando las mismas técnicas sobre placas Basys 3, siendo la máxima frecuencia alcanzada en esta última de 100MHz, para la misma longitud.

El *software* desarrollado para la realización de este TFG se puede encontrar en el siguiente enlace:

<https://drive.google.com/drive/folders/1walhurSwOrJHizj-EG1DWmkHI2sqMHsD> [19]

Conclusions

In order to bombard several FPGAs with high speed neutrons, with a particle accelerator, to study their behavior, and evaluate the quality of the protection techniques used in the designs they implement against the SEUs, the need arose to generate in an external and totally controlled way the clock signal that governs the synchronism in said designs. In this way it is possible to force an expected state in the circuits under test and compare them with those obtained during the experiment. This is how this TFG arises, which is framed in the context described.

When transmitting said signal externally, problems arise in the reception and transmission of this signal in a reliable manner. Due to the electromagnetic and impulsive noise present in these environments, generated largely by the particle accelerator itself, as well as by the power supply of the devices involved, it is necessary to immunize said signal before them. In this TFG this problem has been addressed through the differential transmission of signals and using techniques and elements capable of reducing the noise it originates.

Thanks to the generation of the clock signal in a differential manner, it has been possible to reliably transmit the clock signal between two FPGA Virtex 5 at a frequency of 170MHz, using one of the twisted pairs that constitute a cable of textit UTP Ethernet of category 6 up to 20m in length. This signal has been transmitted through two SMA connectors integrated in a development board XUPV5-LX110T and they have been generated with an external circuit to the FPGA present in the board. Due to the possibility of instantiating a termination resistor present in the board, which avoids the use of external solders and resistors, and the range of voltages with which the differential signal is built, better results have been obtained than using the same techniques on Basys 3 board, with the maximum frequency reached in the latter of 100MHz for the same length.

The software developed for the realization of this work can be found in the following link:

<https://drive.google.com/drive/folders/lwalhurSwOrJHizj-EGlDWmkHI2sqMHsD> [19]

Capítulo 7

Futuras líneas de investigación

Debido a las complicaciones surgidas durante el estudio, derivadas de la mala adaptación entre las impedancias de los circuitos involucrados y la impedancia característica del cable utilizado, sería interesante continuar la investigación en esta línea para intentar aumentar la frecuencia de transmisión. Este estudio deja diferentes líneas por las que se puede continuar la investigación, se nombrarán las más destacables a continuación:

- Estudio de las frecuencias y distancias máximas alcanzables utilizando cables de diferentes impedancias.
- Toma de resultados de la transmisión en entornos de ruido electromagnético e impulsivo elevados.
- Estudio de la transmisión fiable utilizando para ello cables con apantallamiento.
- Estudio con FPGA que posean IOB de tipo HP (*High Performance*), los cuales permiten utilizar estándares de salida cuya impedancia puede ser adaptada.

Bibliografía

- [1] J. Hussein and G. Swift, “Mitigating Single-Event Upsets Xilinx White Paper (wp395)(v1.1),” 2015.
- [2] Diligent, “Basys 3 FPGA Board Reference Manual,” p. 19, 2016.
- [3] Xilinx, “M1505/6/7 Virtex-5 Evaluation Platform,” *ml50xschematics*, Jan, 2008.
- [4] Xilinx, “7series FPGAs Configurable Logic Block,” *UG474*, Sep, p. 74, 2016.
- [5] Xilinx, “series7 FPGAs Packaging and Pinout,” *UG475*, Mar, p. 340, 2018.
- [6] Xilinx, “series7 FPGAs clocking resources,” *UG472*, Jun, vol. 12, p. 114, 2015.
- [7] Xilinx, “Virtex-5 FPGA user guide,” *UG190*, vol. 5, 2009.
- [8] Xilinx, “Microblaze processor reference guide,” *UG081*, 2008.
- [9] Xilinx, “AXI Reference Guide,” *UG761*, Mar, 2011.
- [10] Xilinx, “Clocking Wizard(v5.3),” *PG065*, Oct, 2016.
- [11] Xilinx, “series7 FPGAs SelectIO resources,” *UG471*, Aug, vol. 20, p. 188, 2015.
- [12] W. J. Dally and J. W. Poulton, *Digital systems engineering*. Cambridge University Press, 2008.
- [13] J. S. Patrón, L. V. Díaz, and M. V. Solano, “Diseño e implementación de algoritmo para el procesamiento de imágenes en sistemas embebidos,” *Ingeniería y Región*, 2013.
- [14] Altera, “Introduction to Single-Event Upsets,” p. 10, 2013.
- [15] S. Churiwala, *Designing with Xilinx® FPGAs: Using Vivado*. Springer, 2016.
- [16] M. Meyers, *Network+ Certification: All-in-One Exam Guide*. McGraw-Hill Professional, 2002.
- [17] A. R. C. Lechtaler and R. J. Fusario, *Teleinformática para ingenieros en sistemas de información*, vol. 2. Reverté, 1999.
- [18] Xilinx, “Vivado Design Suite 7series FPGA Libraries Guide,” *UG953*, Jul, p. 416, 2012.

- [19] D. Ortiz, “<https://drive.google.com/drive/folders/1wa1hursworjhizj-eg1dwmkhi2sqmhsd>,” 2018.
- [20] Xilinx, “Vivado Design Suite User Guide,” *UG973, Apr*, p. 67, 2018.